TIROCINIO

ANNO ACCADEMICO 2005/2006

II FACOLTA' DI INGEGNERIA

CORSO DI STUDI IN INGEGNERIA ELETTRONICA

CAGNONE ANDREA MATRICOLA: 119443

MARACICH GABRIELE MATRICOLA: 119456

AZIENDA OSPITANTE: ITI OMAR

INDICE

INDICE 2
INTRODUZIONE 3
ROBOT “POMPIERE” 4
ROBOT “GUARDIANO” 10
POSSIBILI SVILUPPI 15
BASI ROBOT (MMP-5/MMP-8) 16
MINI ATOM BOT BOARD 25
MICROCONTROLLORE BX24 29
OOPic-R 35
SERVI RC 38
SENSORE DI PROSSIMITA’ AD INFRAROSSI (IRPD-V7) 41
SENSORE PIROELETTRICO TPA 81 45
SEGNALATORE DI POSIZIONE INFRAROSSI (BEACON) 50
SENSORE DI DISTANZA AD ULTRASUONI – SRF04 53
VENTOLA 56
SOFTWARE ROBOT “POMPIERE” - BX-24 66
SOFTWARE ROBOT “POMPIERE” - OOPic 74
SOFTWARE ROBOT “GUARDIANO” - BX-24 87
TELECAMERA WIRELESS 93
CONTROLLORE SERVI (SSC-32) 96
BUSSOLA ELETTRONICA 100
CMUCAM2 E CMUCAM2HR E SUPPORTO SU DUE ASSI 104
SINTETIZZATORE VOCALE (SpeakJet) 115

INTRODUZIONE

L'obiettivo del tirocinio è quello di sviluppare alcune applicazioni nell'ambito della robotica. Oltre alla realizzazione è stata sviluppata un'attività di documentazione del lavoro svolto e dei componenti utilizzati.

Le applicazioni da noi realizzate sono due robot semovibili su ruote composti da una piattaforma comune e da kits di schede elettroniche: uno dei due robot, da noi denominato "Pompiere", è in grado di muoversi in un ambiente e rilevare la presenza di un eventuale piccolo incendio e spegnerlo; il secondo robot, denominato "Guardiano", può essere utilizzato nell'ambito della videosorveglianza grazie ad una telecamera wireless onboard.

Prima di passare al montaggio ed alla programmazione dei due robot è stata necessaria una fase di ricerca e studio per capire nel dettaglio il funzionamento dell'hardware a disposizione. Questa fase ha compreso un'attività di traduzione e semplificazione della manualistica presente in modo che gli studenti dell'Istituto Tecnico Industriale fossero in grado di comprenderla ed utilizzarla all'interno dei propri corsi di studi di robotica.

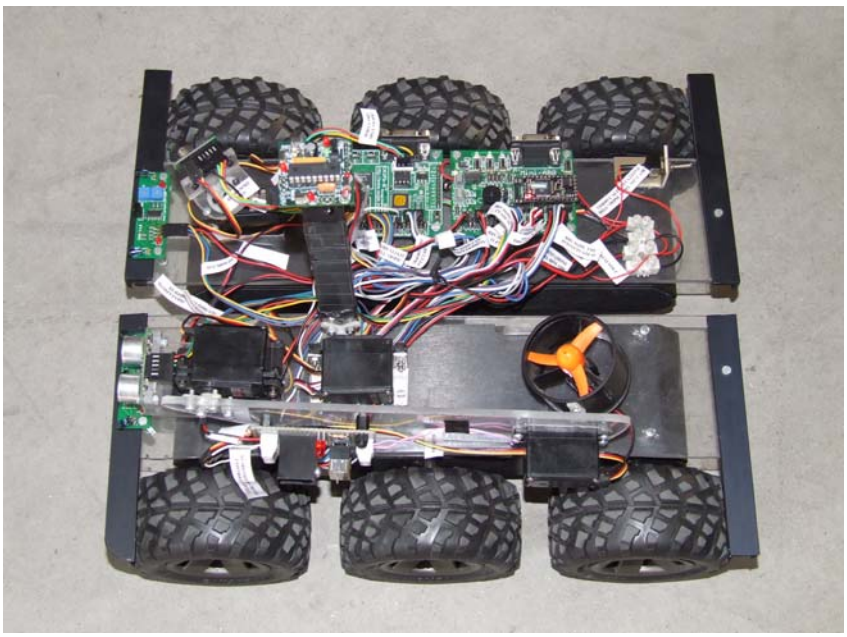
Di seguito sono illustrati i due robot realizzati e alcuni possibili sviluppi con la relativa documentazione.

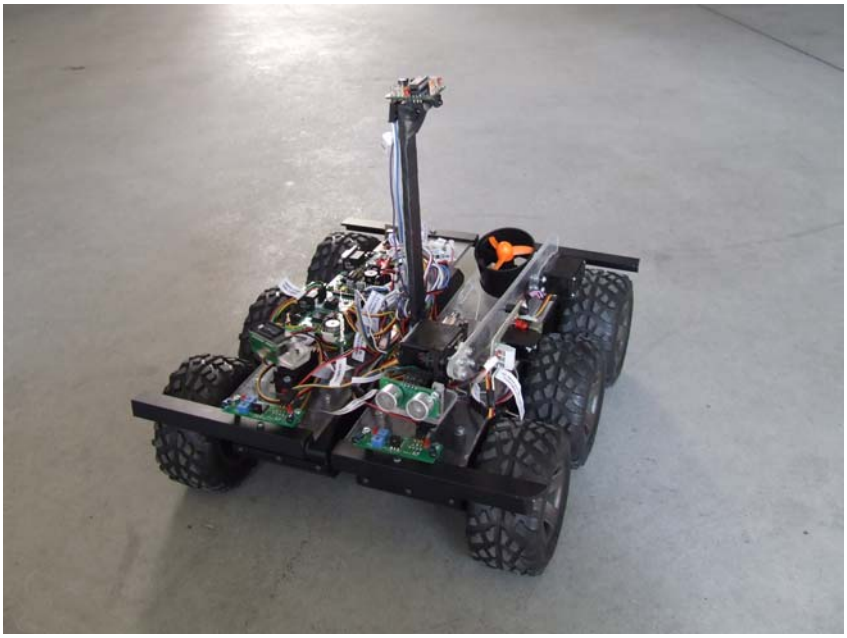
ROBOT “POMPIERE”

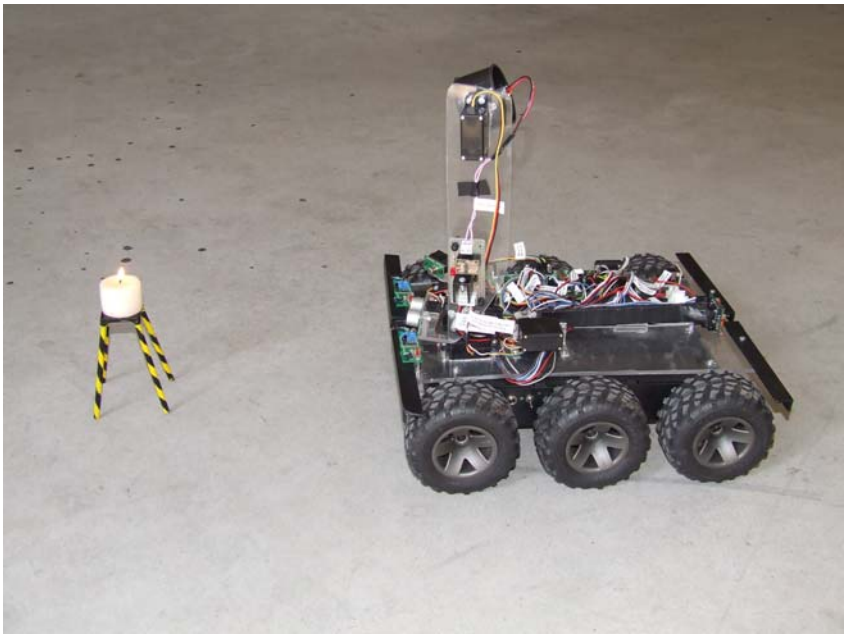
Come già accennato nell’introduzione il robot “Pompiere” è in grado di muoversi autonomamente, cioè senza alcun controllo dall’esterno, evitando i possibili ostacoli per mezzo di sensori di prossimità ad infrarossi. Il movimento è reso possibile dai motori che si trovano all’interno della base.
Sul robot è presente un sensore piroelettrico che ruota in continuazione ed è in grado di rilevare la presenza di fiamme e, nel caso ne vengano rilevate, viene azionata una ventola orientabile montata su braccio che spegne le stesse. Una volta spente le fiamme il robot riprende ad ispezionare l’ambiente e dopo quattro minuti a partire dall’accensione dei circuiti viene attivato il beacon che permette al “Pompiere” di tornare a casa o comunque seguire un altro robot. Tutti i movimenti, quello del sensore piroelettrico, della ventola e del braccio sono possibili grazie all’utilizzo di servocomandi.
Vi è anche montato un rilevatore di distanza ad ultrasuoni che è in grado di rilevare esattamente la distanza dell’oggetto più vicino. Dopo alcuni test ci siamo resi conto che non era indispensabile per il raggiungimento dei nostri obiettivi , dunque l’abbiamo disabilitato. In ogni caso potrebbe essere usato per eventuali usi futuri.
Tutte le operazioni sono coordinate da due microcontrollori: il BX24 e l’OOPIC.

ELENCO COMPONENTI

- Base MMP-8
- Scheda Mini-ABB dove risiede il BX-24
- Microcontrollore BX-24
- Scheda OOPic-R dove risiede l’OOPic
- Cinque servocomandi (rotazione sensore piroelettrico, rotazione braccio-ventola, sollevamento braccio-ventola, orientamento ventola,sollevamento braccio Beacon)
- Due sensori di prossimità ad infrarossi
- Un sensore piroelettrico
- Un segnalatore di posizione (Beacon)
- Un sensore di distanza ad ultrasuoni
- Ventola e relativo controllo da noi realizzato
- Il software del BX-24
- Il software sull’OOPic







CABLAGGIO ROBOT "POMPIERE"

MINI ABB

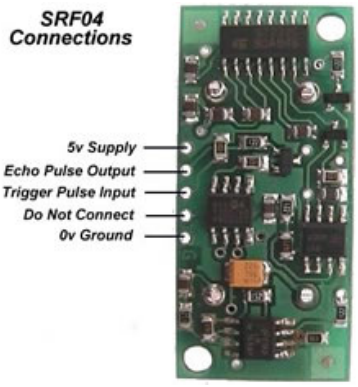
PORTE DA 1 A 3

La porta 1 va collegata al servo che solleva il beacon.

I pin di segnale delle porte 2 e 3 vanno collegati al sensore di distanza ad ultrasuoni rispettivamente:

PIN 2→ Uscita impulso di eco

PIN 3→ Ingresso impulso di trigger



PORTE DA 4 A 7

I pin di segnale delle porte da 4 a 7 vanno collegati ai pin di segnale delle porte da 1 a 4 della scheda OOPic-R rispettivamente:

PIN 4→PIN 4

PIN 5→PIN 3

PIN 6→PIN 2

PIN 7→PIN 1

PORTE 10 E 11

Le porte 10 e 11 sono collegate rispettivamente ai controllori PWM dei motori di sinistra e di destra.

PORTE DA 12 A 15

I pin di segnale delle porte da 12 a 15 vanno collegati alle uscite digitali del beacon rispettivamente:

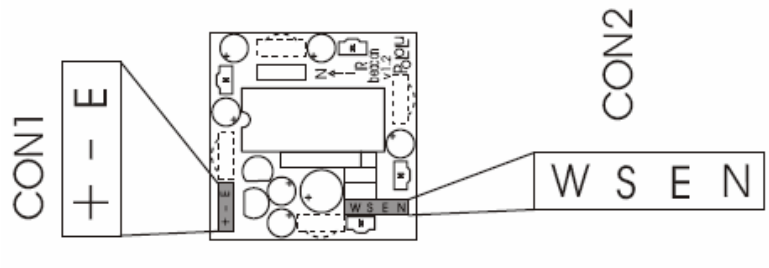
PIN 12→Ovest (W)

PIN 13→Sud (S)

PIN 14→Est (E)

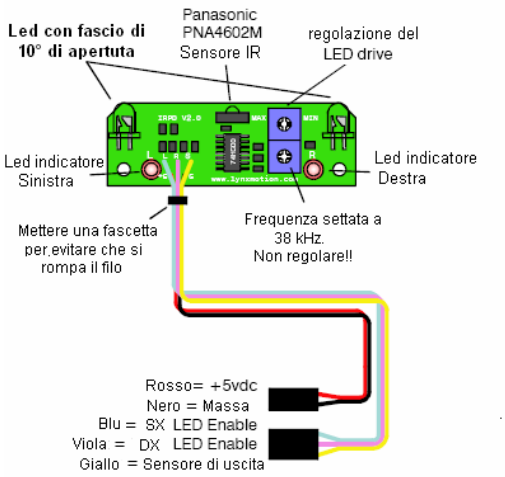
PIN 15→Nord (N)

I pin di alimentazione e massa della porta 12 vanno collegati all'ingresso di alimentazione del beacon (rispettivamente + e -).
I pin di alimentazione e massa della porta 15 vanno collegati all'ingresso di alimentazione del sensore di distanza ad ultrasuoni (rispettivamente 5V e 0V).



PORTE DA AX0 AD AX3

I pin di alimentazione della porta AX0 sono collegati all'ingresso di alimentazione del sensore di prossimità ad infrarossi destra.
I pin di alimentazione della porta AX3 sono collegati all'ingresso di alimentazione del sensore di prossimità ad infrarossi di sinistra.


Led con fascio di 10° di apertuta
Panasonic PNA4602M Sensore IR
regolazione del LED drive
Led indicatore Sinistra
Led indicatore Destra
Mettere una fascetta per.evitare che si rompa il filo
Frequenza settata a 38 kHz. Non regolare!!
Rosso= +5vdc
Nero = Massa
Blu = SX LED Enable
Viola = DX LED Enable
Giallo = Sensore di uscita

CONNETTORI

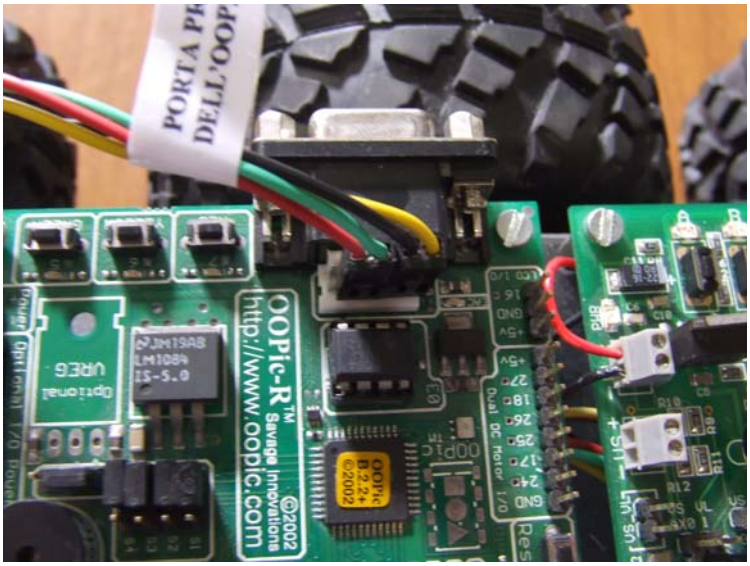
Il connettore VL va collegato alla batteria.

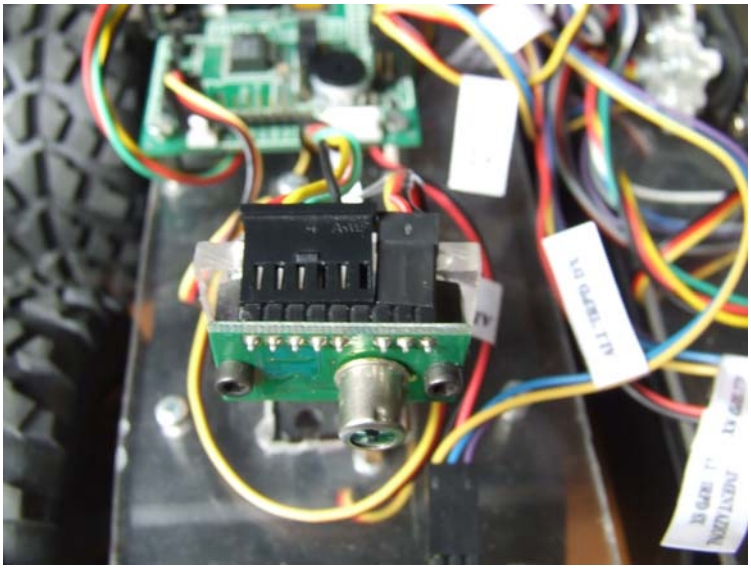
OOPic-R

L'OOPic-R comunica con la Mini-ABB attraverso le porte da 1 a 4.

CONNETTORE Prg

Il connettore Prg è collegato al sensore piroelettrico nel modo indicato in figura.





PORTA 8

Il pin di segnale della porta 8 va collegato all'ingresso di controllo del circuito che pilota la ventola.

PORTE DA 10 A 12

I pin di segnale delle porte da 10 a 12 sono collegate al sensore di prossimità ad infrarossi di destra rispettivamente:

PIN 10 → Enable LED di sinistra

PIN 11 → Enable LED di destra
PIN 12 → Uscita sensore

PORTE DA 13 A 15

I pin di segnale delle porte da 10 a 12 sono collegate al sensore di prossimità ad infrarossi di destra rispettivamente:
PIN 13 → Enable LED di sinistra
PIN 14 → Enable LED di destra
PIN 15 → Uscita sensore

PORTA DA 29 A 31

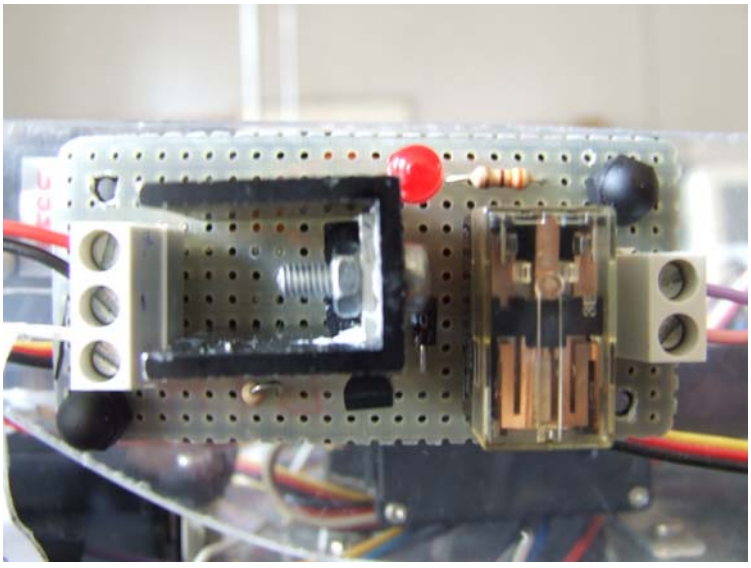
La porta 29 è collegata al servo che controlla l'orientamento della ventola.
La porta 30 è collegata al servo che fa ruotare il braccio su cui è montata la ventola.
La porta 31 è collegata al servo che fa sollevare il braccio su cui è montata la ventola.

CONNETTORE ALIMENTAZIONE

Il connettore di alimentazione è collegato alla batteria.

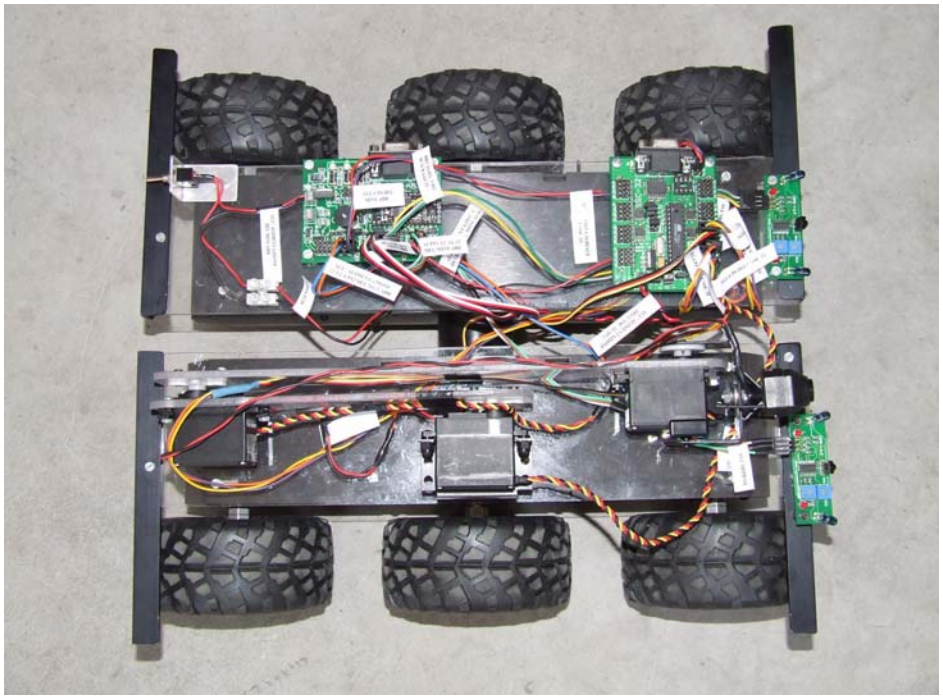
CIRCUITO DI CONTROLLO VENTOLA

La morsettiera d'ingresso del circuito va cablata nel seguente modo:
FILO BIANCO →Pin di segnale della porta 8 dell'OOPic-R
FILO ROSSO →Positivo alimentazione batteria
FILO NERO →Negativo alimentazione batteria



La morsettiera d'uscita del circuito va collegato alla ventola rispettivamente:
FILO ROSA →Positivo della ventola
FILO VIOLA →Negativo della ventola

ROBOT “GUARDIANO”

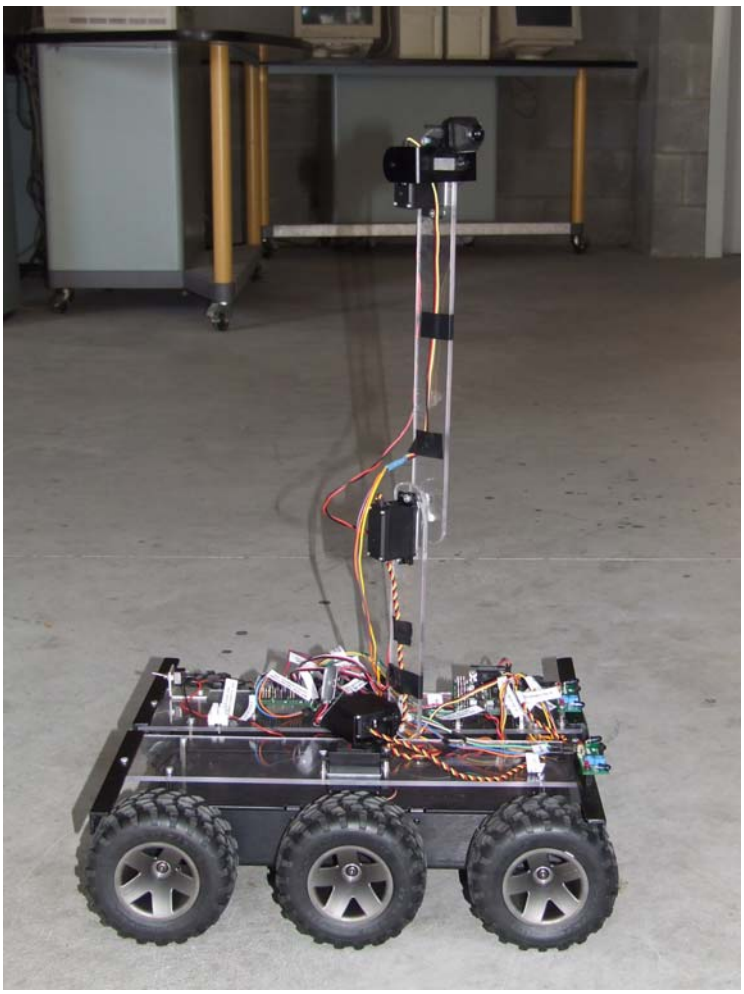


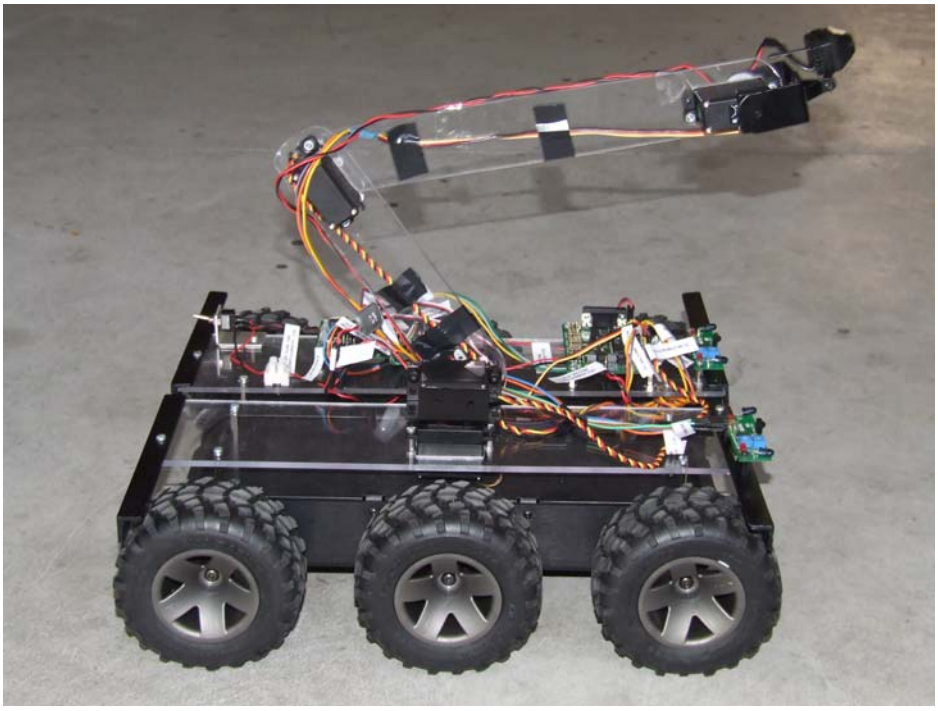
Questo robot, come il precedente ,è in grado di muoversi autonomamente, cioè senza alcun controllo dall’esterno, evitando i possibili ostacoli per mezzo di due sensori di prossimità ad infrarossi: essi sono posizionati nella parte anteriore in modo tale che per rilevare oggetti alla destra del robot viene utilizzato solo il sensore di destra, viceversa per la sinistra, mentre per rilevare ostacoli al centro vengono usati entrambi.
Il movimento è reso possibile dai motori che si trovano all’interno della base.
Sul robot è presente una videocamera wireless montata su un braccio rotante snodato che permette al robot di sorvegliare l’ambiente in cui si trova e di inviare le immagini riprese ad un ricevitore che può trovarsi anche ad una distanza di diversi metri.. Il movimento del braccio è pilotato dal microprocessore BX-24 a cui è connesso un apposito circuito che consente il controllo di più servi in modo coordinato. Tra le possibili modalità di funzionamento abbiamo scelto di fermare il robot e azionare braccio ad intervalli di un minuto permettendo, in tali frangenti, un controllo più ampio dell’ambiente circostante, per riprendere poi il normale funzionamento. Essendo la telecamera montata sulla punta del braccio e dovendo i servi sollevare il peso dello stesso si è reso necessario utilizzare servi con coppia maggiore (HS-635HB) per i due snodi del braccio rispetto agli altri due servi (HS-422) che non sono sottoposti ad uno sforzo così gravoso.
Nei servocomandi la coppia è funzione dell’alimentazione (MAX 6V), maggiore è quest’ultima maggiore è la coppia: abbiamo quindi optato per l’utilizzo di un’ ulteriore batteria composta da 5 celle (1,2V e 2600mAH NiMh) che alimenta da sola l’SSC-32 e

quindi tutti i servi del braccio in alternativa all'utilizzo della tensione a 5V fornita dalla scheda Mini-ABB.
Un'altra ragione per cui abbiamo deciso l'introduzione di questa batteria era evitare di surriscaldare e bruciare il regolatore di tensione sul Mini-ABB che non è in grado di erogare una corrente sufficiente a pilotare i servi.
Tutte le operazioni sono coordinate dal microcontrollore BX24.

ELENCO COMPONENTI

- Base MMP-8
- Scheda Mini-ABB dove risiede il BX-24
- Microcontrollore BX-24
- Quattro servocomandi (rotazione braccio,azionamento dei due snodi del braccio,inclinazione videocamera)
- Due sensori di prossimità ad infrarossi
- Telecamera con trasmettitore wireless
- Controllore servi SSC-32
- Pacco batterie aggiuntivo da 2600mAH Sanyo
- Il software del BX-24





CABLAGGIO ROBOT "GUARDIANO"

MINI-ABB

PORTA 1

I pin di alimentazione della porta 1 vanno collegati agli ingressi di alimentazione del sensore IRPD di destra.

PORTA 3

Il pin di segnale e quello di massa della porta 3 vanno collegati alla porta seriale dell'SSC-32.

PORTE 6 E 7

Le porte 6 e 7 sono collegate rispettivamente ai controllori PWM dei motori di sinistra e di destra.

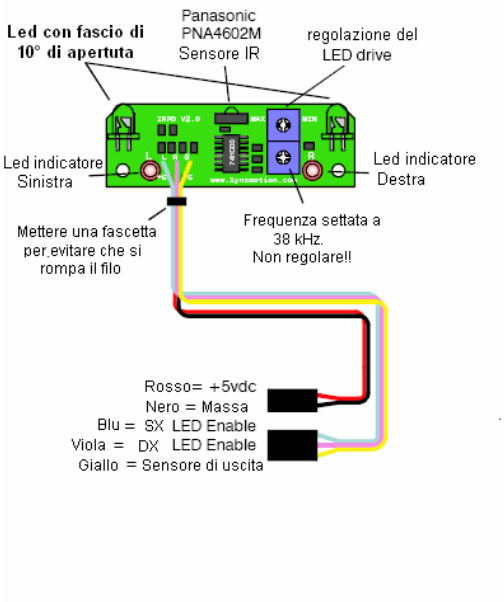
PORTE DA 8 A 10

I pin di segnale delle porte da 8 a 10 sono collegate al sensore di prossimità ad infrarossi di sinistra rispettivamente:

PIN 8 → Enable LED di sinistra

PIN 9 → Enable LED di destra

PIN 10 → Uscita sensore


Led con fascio di 10° di apertuta
Panasonic PNA4602M Sensore IR
regolazione del LED drive
Led indicatore Sinistra
Led indicatore Destra
Mettere una fascetta per.evitare che si rompa il filo
Frequenza settata a 38 kHz. Non regolare!!
Rosso= +5vdc
Nero = Massa
Blu = SX LED Enable
Viola = DX LED Enable
Giallo = Sensore di uscita

PORTA 12

I pin di alimentazione della porta 12 vanno collegati agli ingressi di alimentazione del sensore IRPD di sinistra.

PORTE DA 13 A 15

I pin di segnale delle porte da 13 a 15 sono collegate al sensore di prossimità ad infrarossi di destra rispettivamente:

PIN 15 → Enable LED di sinistra

PIN 14 → Enable LED di destra

PIN 13 → Uscita sensore

SSC-32

L'SSC-32 comunica con la Mini-ABB attraverso la porta seriale.

ALIMENTAZIONE

Il connettore di alimentazione VS2 va collegato al pacco batterie aggiuntivo.

PORTA 0

La porta 0 va collegata al servo che controlla l'inclinazione della parte inferiore del braccio.

PORTA 3

La porta 3 va collegata al servo che controlla l'inclinazione della parte superiore del braccio.

PORTA 4

La porta 4 va collegata al servo che controlla l'orientamento della telecamera.

PORTA 6

La porta 6 va collegata al servo che controlla la rotazione del braccio.

POSSIBILI SVILUPPI

Oltre alle due configurazioni descritte sopra, sulle basi MMP-5 e MMP-8, con un po' di fantasia e con alcune modifiche hardware e software, è possibile realizzare altre applicazioni utilizzando il materiale a disposizione; ad esempio i sensori ad infrarossi che permettono di evitare gli ostacoli possono essere sostituiti con i sensori ad ultrasuoni oppure il braccio del "Guardiano", invece di essere pilotato tramite BX-24, può essere controllato in remoto con un radiocomando o ancora può essere utilizzata una bussola elettronica che rileva la posizione esatta del robot consentendogli di seguire un percorso prestabilito.. Nel nostro caso abbiamo utilizzato la base MMP-8 perché, pur essendo pressoché equivalente nelle prestazioni e nel funzionamento alla MMP-5, ha dimensioni maggiori il che ci permette di integrare un maggior numero di kit, e quindi di applicazioni,su uno stesso robot. Un'altra funzione che è possibile aggiungere è quella di rendere il robot parlante inserendo uno speaker con relativa scheda di controllo, la quale permette di memorizzare vari messaggi associati a diversi eventi che possono occorrere (ad esempio il "Pompiere" può emettere un messaggio d'allarme quando rileva una fiamma oppure segnalare la presenza di eventuali ostacoli). Uno degli sviluppi più interessanti applicabili a questi robot è quello di utilizzare una videocamera la quale, opportunamente interfacciata con il microcontrollore, consente agli stessi di inseguire degli oggetti o di rilevare dei movimenti.
Di seguito è riportato un elenco dell'hardware che non è stato inserito nella versione finale dei due robot, ma che comunque è stato da noi testato e documentato.

- Bussola elettronica
- Videocamera CMUcam2
- Altoparlante e sintetizzatore vocale

BASI ROBOT (MMP-5/MMP-8)

MMP-5

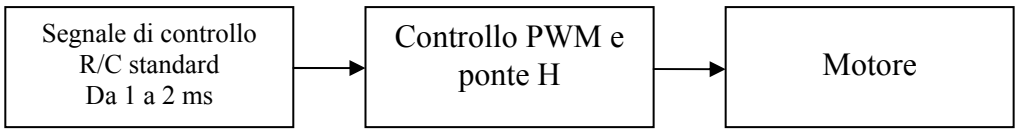
Trazione	4 Ruote motrici (pneumatici Geolanders)
Telaio	Telaio in alluminio a 2 pezzi
Motori	Motori 4 x 12VDC con riduttore 30:1
Velocità	Circa 3,84 km/h
Controllo motori	2 x 12 Amp Controllori PWM Segnale (R/C Std.) 1.0-2.0 mS Controllo seriale opzionale
Batterie	12V 1400 mAh NiMH (due pacchi da 5 celle)
Durata batterie	1 ora +
Peso	2.04kg
Carico utile	1.8kg
Altezza	10.4cm
Lunghezza	26cm
Larghezza	26cm

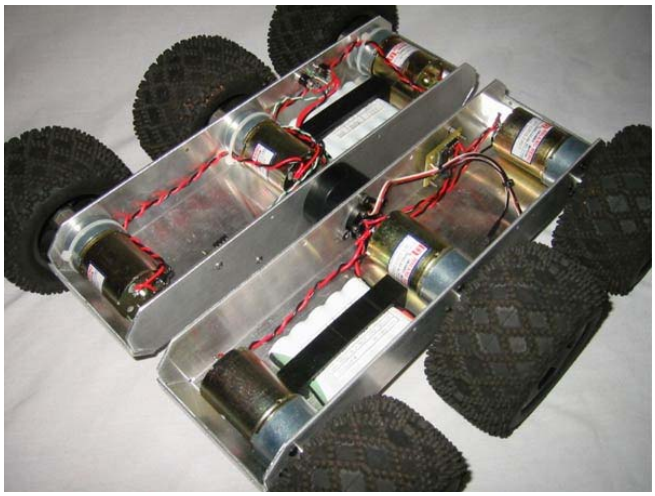
MMP-8

Trazione	6 Ruote motrici (pneumatici Geolanders)
Telaio	Telaio in alluminio a 4 pezzi
Centro di rotazione passivo	Angolo do rotazione di 45 gradi
Motori	Motori 6 x 12VDC con riduttore 30:1
Velocità	Circa 3,84 km/h
Controllo motori	2 x 12 Amp Controllori PWM Segnale (R/C Std.) 1.0-2.0 mS Controllo seriale opzionale
Batterie	12V 1400 mAh NiMH (due pacchi da 5 celle)
Durata batterie	45 min +
Peso	3.40
Carico utile	3.18kg
Altezza	10.41cm
Lunghezza	35.56cm
Larghezza	31.65cm

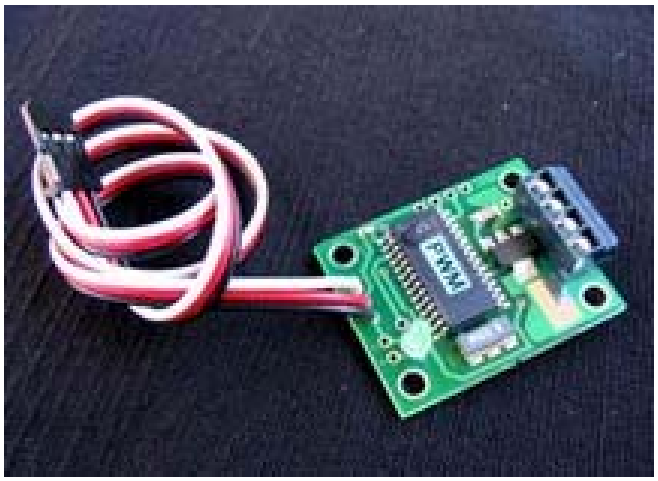
Queste piattaforme sono state progettate per avere una grande versatilità e possono quindi essere utilizzate in diversi campi: educativo, ricerca e sviluppo, sorveglianza e sicurezza, esplorazione, lavori rischiosi e intrattenimento. L' MMP-8, rispetto all'MMP-5, possiede uno snodo centrale che permette un movimento parzialmente indipendente delle parti destra e sinistra del robot. Le piattaforme MMP sono dotate di controlli della velocità elettronici di tipo PWM (Pulse Width Modulation)che utilizzano il segnale di controllo R/C standard. Il

segnale è composto da un impulso la cui durata spazia da 1mS a 2mS: questo significa che variando la durata dello stesso si possono controllare i motori di destra e di sinistra raggiungendo la massima velocità, in avanti (1mS) e all'indietro (2mS), oppure fermandoli (1,5mS). Riducendo la velocità dei motori di destra o di sinistra e mantenendo costante la velocità dei motori sul lato opposto il robot curverà rispettivamente a destra o a sinistra: la velocità di tutti i motori di uno stesso lato è uguale ed è controllata da un singolo circuito PWM.

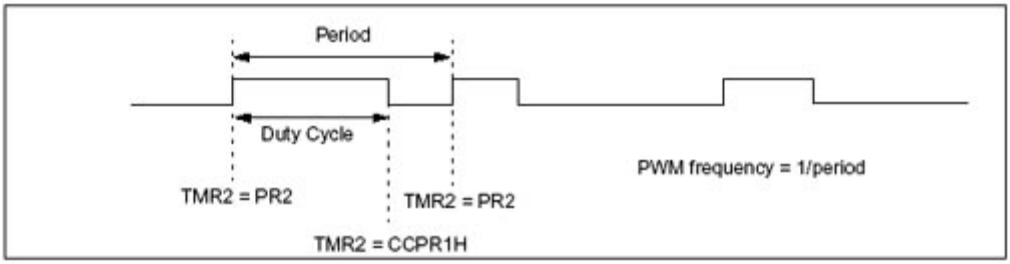




CONTROLLO PWM

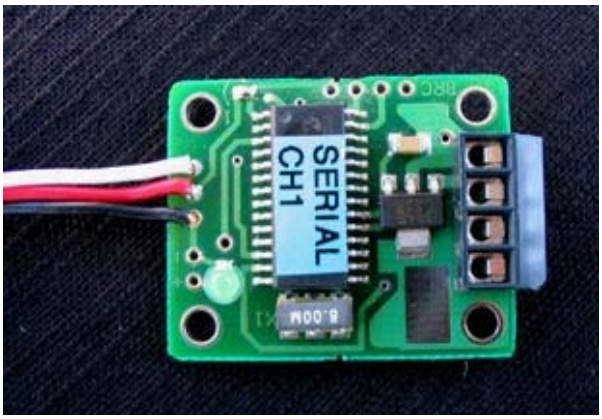


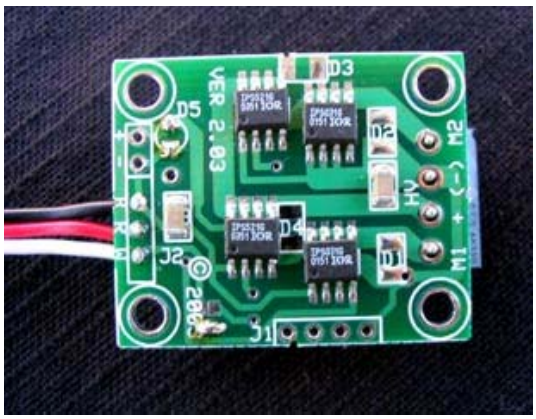
Ciascun circuito converte il segnale ricevuto dal microcontrollore (da 1mS a 2mS) in un segnale PWM. Un segnale Pulse Width Modulation (ovvero modulazione a variazione della larghezza d'impulso) è un' onda quadra di duty cycle variabile che permette di controllare l'assorbimento (la potenza assorbita) di un carico elettrico(nel nostro caso il motore DC), variando (modulando) il duty cycle.



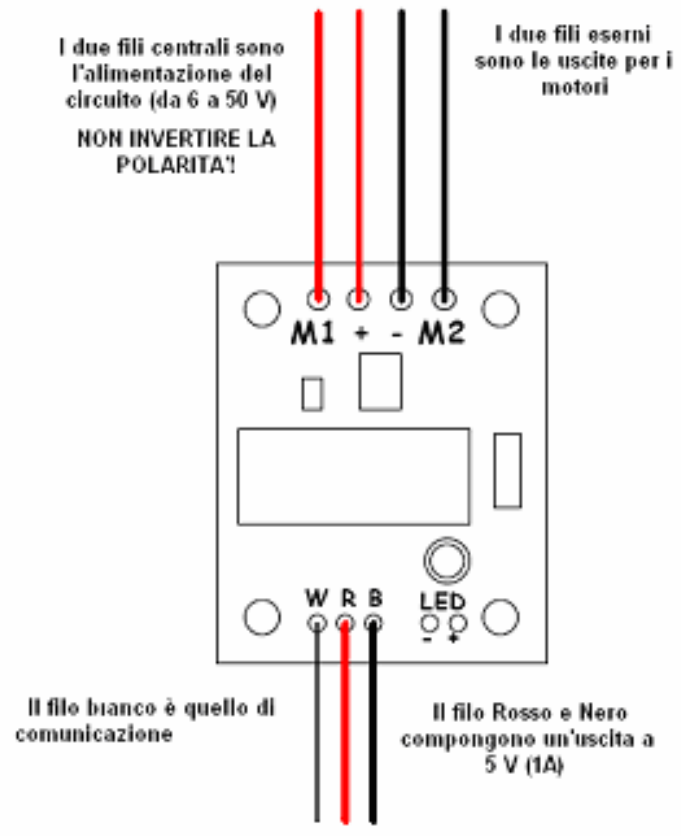
Un segnale PWM è caratterizzato da frequenza (fissa) e duty cycle (variabile); come si deduce dalla figura, il duty cycle è il rapporto tra il tempo in cui l'onda assume valore alto e il periodo T (l'inverso della frequenza: T=1/f).Questo tipo di controllo permette solo la regolazione della velocità; per invertire il regime di rotazione si utilizza il ponte H.

CONTROLLO DEI MOTORI SERIALE





Questo circuito serve per pilotare motori con un segnale seriale; ai motori comunque arriva un segnale di tipo PWM.


I due fili centrali sono l'alimentazione del circuito (da 6 a 50 V)
NON INVERTIRE LA POLARITA'!
I due fili esterni sono le uscite per i motori
M1 + - M2
W R B
LED
- +
Il filo bianco è quello di comunicazione
Il filo Rosso e Nero compongono un'uscita a 5 V (1A)

Caratteristiche:

- Singolo canale con comando proporzionale (avanti e indietro)
- Led di stato
- 10A regolati (12A di picco)
- Limitatore di corrente
- Protezione termica

- Tensione di ingresso da 6 a 50 Volts
- Uscita a 5 Volt regolati per poter eventualmente alimentare una ricevente R/C (circa 1A)
- Connettore Futaba di tipo J
- Morsettiera per collegare l'alimentazione e l'uscita motori
- Pilotaggio attraverso interfaccia RS-232

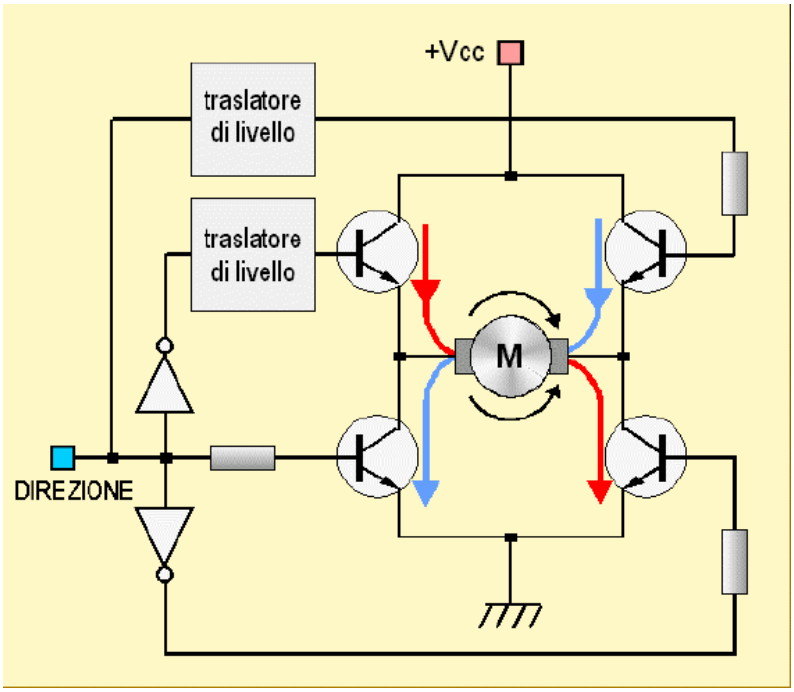
Funzionamento:

La velocità della porta seriale deve essere settata a 9600 bit per secondo, 8 bit dati e uno di stop.
Il protocollo è a 2 Byte, il primo è il canale e il secondo i dati con il seguente formato:

BYTE 1: 1110xxxx BYTE 2: 0xxxxxxx

Il byte dei dati può contenere valori da 0 a 63 per far ruotare il motore in un senso e da 64 a 127 per il verso opposto. Variando il valore si varia la velocità.

PONTE AD H



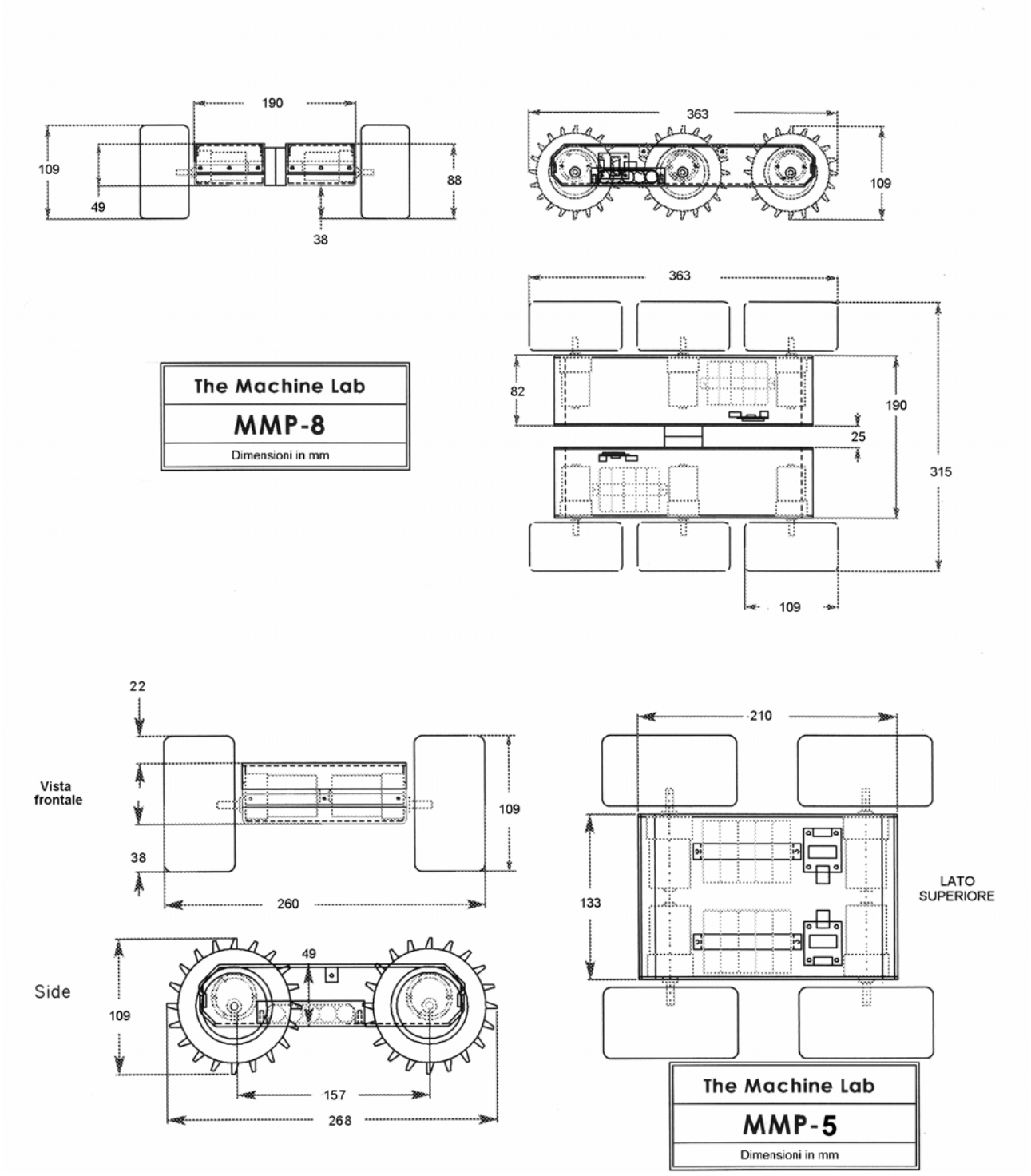
Per far girare i motori in entrambi i versi è necessario invertire il segno della corrente che passa all'interno dei motori stessi, per far ciò si usa un circuito chiamato ponte H costituito da 4 interruttori comandati dal segnale di direzione. Quando il livello del segnale di direzione è a livello alto i transistor (che possono essere anche MOSFET) attraversati dalla corrente, indicata in figura dalla linea blu, portano in rotazione il motore in una direzione, mentre quando il livello del segnale è basso vengono attivati solo gli altri due transistor (attraversati dalla linea rossa) facendo ruotare il motore nella direzione opposta .

MOTORI

All'interno delle due basi sono presenti gli stessi motori. Sono motori a 12 V con ingranaggi e riduttori 30:1 metallici. Di seguito sono riportate le caratteristiche degli stessi.

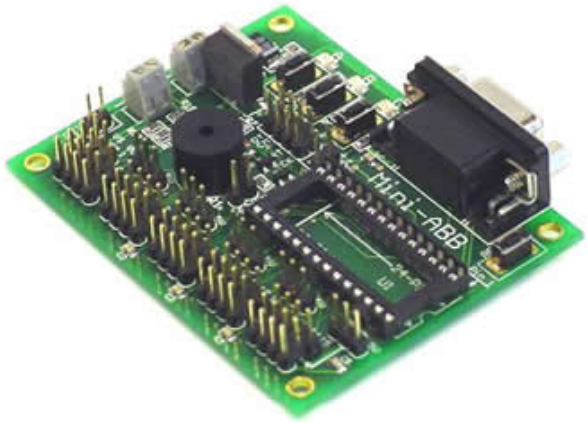
Peso: 206g
200 RPM (giri al minuto)
Coppia 2,2 kg*cm
Efficienza del 78%

DISEGNI QUOTATI



Vai a Robot “Pompiere”
Vai a Robot “Guardiano”
Vai a Possibili sviluppi
Torna all’indice

MINI ATOM BOT BOARD



La scheda Mini-ABB e' compatibile con tutti i microcontrollori stamp-like a 24 e 28 pin, possono quindi essere installati: Basic ATOM 24 e 28pin, Basic ATOM PRO, BasicX BX24, OOPic-C, Basic Stamp 24 e 28pin.
Tutti gli ingressi e le uscite sono comodamente riportate e molte hanno piu' funzioni, selezionabili con dei comodi jumpers a seconda dell'utilizzo che si vuole assegnargli. C'e' un piccolo altoparlante piezoelettrico incorporato, con relativo transistor di amplificazione, tre pulsanti e tre LED, una presa per il controller PS2 (richiede adattatore non fornito), un pulsante di reset, ingressi per alimentazione della logica e dei servi, ponticellabile, regolatore da 250mA on board.

CONTROLLARE I SERVI

Questa scheda e' stata progettata per esaltare le doti del microcontrollore Basic ATOM PRO il quale ha il nuovo comando HSERVO. Esso puo' pilotare i servocomandi in background. Equivale ad un servo controller vero e proprio, si puo' controllare posizione e velocita' dei servocomandi cambiando semplicemente una variabile, e' come avere un servo controller e un microcontrollore sulla stessa scheda.

CARATTERISTICHE PRINCIPALI

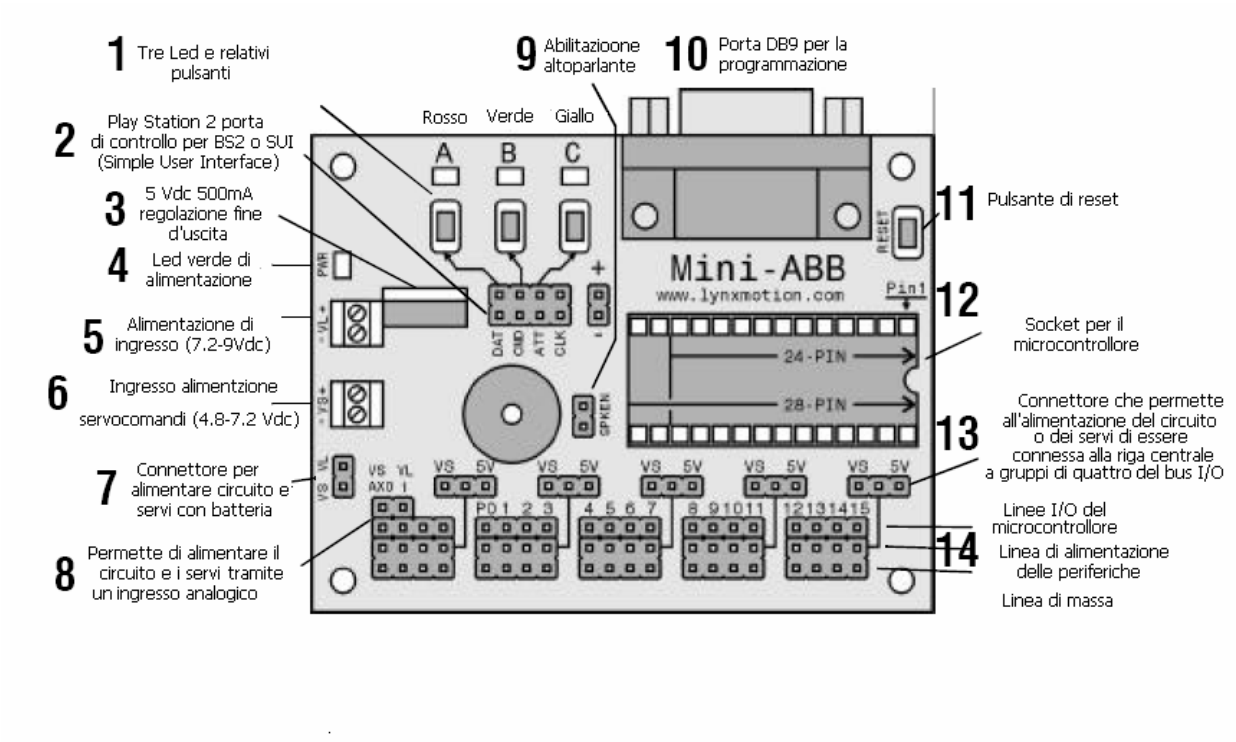
Altoparlante : un piccolo altoparlante piezoelettrico incorporato, con relativo transistor di amplificazione
Interfaccia utente: sono stati inseriti 3 pulsanti e 3 LED colorati su 3 porte I/O
Porta PS2 :tramite l'apposito adattatore, e' possibile connettere questo ottimo controller con il quale pilotare robot e bracci robotici, anche senza fili!

Alimentazione: la scheda ha 2 ingressi per l'alimentazione. Uno per il microcontroller e uno per i servocomandi. E' possibile ponticellare le alimentazioni per alimentare sia il micro che i servi con un'unica batteria

Ingressi analogici: inserendo alcuni jumpers e' possibile dirigere gli ingressi di alimentazione della scheda verso gli ingressi A/D del micro per monitorare lo stato delle batterie

Bus I/O: il bus di I/O e' progettato in gruppi di 4, con una fila di pin I/O, una fila di alimentazione e una fila di massa. La fila di alimentazione e' selezionabile tramite jumper, puo' prendere l'alimentazione dal ramo dei servi o da quello della logica.

Plug and Play:la Mini-ABB ha i terminali dell'alimentazione con viti, non c'e' bisogno di saldare i cavi.


1 Tre Led e relativi pulsanti
2 Play Station 2 porta di controllo per BS2 o SUI (Simple User Interface)
3 5 Vdc 500mA regolazione fine d'uscita
4 Led verde di alimentazione
5 Alimentazione di ingresso (7.2-9Vdc)
6 Ingresso alimentzione servocomandi (4.8-7.2 Vdc)
7 Connettore per alimentare circuito e servi con batteria
8 Permette di alimentare il circuito e i servi tramite un ingresso analogico
9 Abilitazioone altoparlante
10 Porta DB9 per la programmazione
11 Pulsante di reset
12 Socket per il microcontrollore
13 Connettore che permette all'alimentazione del circuito o dei servi di essere connessa alla riga centrale a gruppi di quattro del bus I/O
14 Linee I/O del microcontrollore
Linea di alimentazione delle periferiche
Linea di massa
Rosso Verde Giallo
A B C
Mini-ABB
www.lynxmotion.com
24-PIN
28-PIN

1) I 3 LED e pulsanti consentono di creare una semplice interfaccia utente per il programma.

Chip I/O	PB/LED
Pin 4	A RED
Pin 5	B GREEN
Pin 6	C YELLOW

2) Porta utilizzata per connettere il controller di una Playstation Sony ed utilizzarlo come controller del robot: essa è progettata specificatamente per il microcontrollore Basic Stamp 2. Quando si utilizza questa porta connettere il cavo della PS2 come indicato sotto.

Chip I/O	PS2 cable
Pin 4	DAT Brown
Pin 5	CMD Orange
Pin 6	ATT Blue
Pin 7	CLK Black
*SrvPwr	+7.2 Green
5vdc	+ Yellow
GND	− Black

*Necessario per abilitare le vibrazioni nel controller Sony o per alcune unità wireless.

3) Il regolatore a bassa caduta di tensione permette di fornire i 5vdc necessari al circuito anche avendo solo 5,5vdc in ingresso (tensione di ingresso max 9VDC): esso è in grado di erogare fino a 500mA ma, per valori di correnti così elevati, è consigliabile collegarlo ad un dissipatore per prevenire il surriscaldamento dello stesso.

4) LED alimentazione: si accende quando l'alimentazione è applicata correttamente

5) E' l'ingresso di alimentazione delle logiche ed è detto Logic Voltage,
o VL.: esso è mormalmente connesso ad una batteria a 9vdc per alimentare l' IC Ibrido e qualunque cosa sia collegata alle linee dei 5vdc sulla basetta. Inoltre questo ingresso è utilizzato per isolare l'alimentazione delle logiche da quella dei servi.

Board	Input
VL+	RED
VL-	BLACK

6) E' l'ingresso di alimentazione dei motori dei servi, detto anche VS. (la tensione in ingresso varia tra 4,8vdc e 7,2vdc); VS può alimentare i servi soltanto oppure sia i servi che le logiche.

7) Permette di alimentare i servi e le logiche con la stessa batteria. Connette semplicemente l'input VS con l'input VL. Fare attenzione a non utilizzare l'input VL quando si utilizza questa opzione.

8) Il connettore permette, se si utilizza un microcontrollore Atom-28, di collegare gli ingressi VL e/o VS a due degli ingressi analogici di quest'ultimo.

9) Il jumper abilita lo speaker sulla scheda. Per utilizzare l'altoparlante bisogna inviare il comando di generazione suono appropriato al pin 9.

10) Inserire nel connettore un cavo DB9 e collegarlo ad una porta seriale a 9 pin sul PC per scaricare programmi e ricevere informazioni di debug.

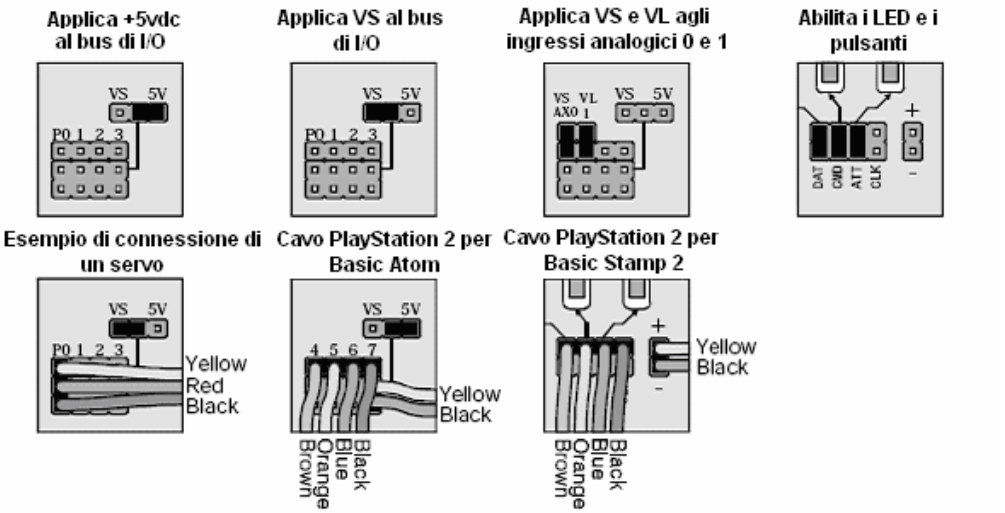
11) E' pulsante serve per resettare il microcontrollore. Può essere utile per far partire un diverso programma a seconda di quale pulsante viene premuto.

12) Zoccolo per montare il microcontrollore.

13) Connettore con il quale si configura la riga centrale del bus di I/O affinché utilizzi VL (+5vdc dal regolatore on board) o VS (direttamente dall'alimentazione dei servi). Ciò viene implementato in gruppi di 4 pin di I/O. Attenzione: applicando l'alimentazione dei servi a questa riga con una periferica a 5vdc installata si danneggia la periferica.

14) Sono le porte con cui si collegano i controlli dei servo motori, i sensori, ecc. al microcontrollore. Fare attenzione quando si connette qualunque cosa al bus di I/O; mai collegare qualcosa quando il circuito è alimentato.

PANORAMICA JUMPER E CONNETTORI



Manuale in inglese:
http://www.lynxmotion.com/images/data/abbpdf02.pdf

Vai a Robot "Pompiere"
Vai a Robot "Guardiano"
Vai a Possibili sviluppi
Torna all'indice

MICROCONTROLLORE BX24

Il BX24 è il piu' diffuso tra i microcontrollori della serie BasicX. In un package DIP24 di 34 x 17 mm e' racchiuso un sistema completo a microcontrollore, programmabile in Basic. Il BX24 ha tutto quello che occorre per il suo funzionamento sulla propria schedina, non necessita di componenti esterni, convertitori seriali o programmatori; il software (IDE) e' gratuito e si scarica direttamente dal sito del produttore utilizzando i seguenti link.
Si programma collegandolo direttamente alla porta seriale del PC. Il linguaggio di programmazione, Basic Express, e' praticamente identico al Visual Basic, questo rende abbastanza semplice la programmazione, creare applicazioni, anche complesse, con il BX24, e' un lavoro di poche ore.
La dotazione hardware di questo chip comprende: Real Time Clock incorporato, 8 porte ADC a 10bit, 2 porte seriali, 2 PWM hardware, interrupt, porta SPI e I2C, convertitore RS232, regolatore di tensione.

- Basic Express 2.10 Setup - Completo Linguaggio di programmazione completo di documentazione (21MB)
- Basic Express 2.10 Setup - Solo Programma Linguaggio di programmazione senza documentazione (4MB)
- Basic Express 2.10 Setup - Documentazione Linguaggio di programmazione solo documentazione (18MB)

BasicX-24 Specifiche tecniche	
Velocita'	83,000 istruzioni al secondo
EEPROM	32K bytes (Programma utente e Dati)
Max lungh. Programma	8000+ linee di linguaggio Basic
RAM	400 bytes
Porte I/O disponibili	21 (16 standard + 2 seriali + 3 con accesso al di fuori dei piedini DIP standard)
Ingressi Analogici (ADCs)	8 (8 delle 16 porte standard I/O possono individualmente funzionare come porte standard digitali oppure come ADC a 10bit)
Freq. di campionamento ADC	6.000 /s max
Velocita della seriale I/O	1200 - 460.8K Baud
Tensione operativa	4.8/15.0 VDC
Temperatura operativa	0°C - +70°C
Corrente necessaria	20mA + il carico sulle porte I/O
Calcoli in Virgola Mobile	Si
Multitasking	Si
Orologio interno	Si
Interfaccia SPI	Si
Interfaccia di programmazione	Seriale ad alta velocita' o parallela.
Package	24 pin DIP
Segnalazioni	2 LED rosso e verde on board programmabili

Nei robot da noi costruiti il Microcontrollore BX24 costituisce il cuore del sistema in quanto è in grado di gestire sensori, servi e motori. Esso è montato su una scheda: il Mini-ABB.

ELENCO DELLE FUNZIONI DISPONIBILI NELL'AMBIENTE DI SVILUPPO:

Funzioni matematiche:

Abs Valore assoluto
ACos Arco coseno
ASin Arco seno
Atn Arco tangente
Cos Coseno
Exp Eleva e ad una specifica potenza
Exp10 Eleva 10 ad una specifica potenza
Fix Tronca un valore floating point
Log Logaritmo naturale
Log10 Logaritmo base 10
Pow Eleva un operando ad una specifica potenza
Randomize Imposta la base di del generatore di numeri casuali
Rnd Generat un numero casuale
Sin Seno
Sqr Radice quadrata
Tan Tangente

Funzioni sulle stringhe

Asc Restituisce il codice ASCII di un carattere
Chr Converte un valore numerico in un carattere
LCase Converte una stringa di caratteri da maiuscoli a minuscoli
Len Restituisce la lunghezza di una stringa
Mid Copia una sottostringa
Trim Ordina gli spazi bianchi di una stringa
UCase Converte una stringa di caratteri da minuscoli a maiuscoli

Funzioni di gestione della memoria

BlockMove Copia un blocco di dati da una locazione all'altra della RAM
FlipBits Genera una copia di una sequenza di bit
GetBit Legge un singolo bit da una variabile
GetEEPROM Legge dati dalla EEPROM
MemAddress Restituisce l'indirizzo di una variabile o di un vettore

MemAddressU Restituisce l'indirizzo di una variabile o di un vettore
PersistentPeek Legge un byte dalla EEPROM
PersistentPoke Scrive un byte nella EEPROM
PutBit Scrive un singolo bit in una variabile
PutEEPROM Scrive dati nella EEPROM
RAMPeek Legge un byte dalla RAM
RAMPoke Scrive un byte nella RAM
SerialNumber Restituisce la versione del chip BasicX

Funzioni di gestione delle code
GetQueue Legge dati da una coda
OpenQueue Definisce un vettore come una coda
PeekQueue Legge i dati da una coda senza rimuoverli
PutQueue Scrive dati in una coda
PutQueueStr Scrive una stringa in una coda
StatusQueue Stabilisce se una coda ha dati disponibili per la lettura

Funzioni di gestione dei task
CallTask Avvia un task
CPUSleep Mette il processore in diverse modalità a basso consumo
Delay Interrompe temporaneamente un task e permette l'esecuzione di un altro task
DelayUntilClockTick Mette in pausa un task fino al successivo colpo di clock
FirstTime Deternima se il programma è mai stato eseguito da quando è stato scaricato
LockTask Blocca il task e impedisce l'esecuzione di altri task
OpenWatchdog Avvia il timer del watchdog
ResetProcessor Resetta e riavvia il processore
Semaphore Coordina la condivisione di dati tra task
TaskIsLocked Determina se un task è bloccato
UnlockTask Sblocca un task
WaitForInterrupt Consente ad un task di rispondere ad un interrupt hardware
Watchdog Resetta il timer watchdog

Funzioni per la conversioni tra tipi di dati
CBool Converte da Byte a Booleano
CByte Converte in Byte
CInt Converte in intero
CLng Converte in LONG

CSng Converte in floating point (singolo)
CStr Converte in una stringa
CuInt Converte in un intero senza segno
CuLng Converte in LONG senza segno
FixB Tronca un valore floating point e lo converte in Byte
FixI Tronca un valore floating point e lo converte in intero
FixL Tronca un valore floating point e lo converte in Long
FixUI Tronca un valore floating point e lo converte in un intero senza segno
FixUL Tronca un valore floating point e lo converte in Long senza segno
ValueS Converte da stringa a floating point (singolo)

Funzioni per la gestione dell'orologio di sistema
GetDate Restituisce la data
GetDayOfWeek Restituisce il giorno della settimana
GetTime Restituisce l'ora
GetTimestamp Restituisce data e giorno
PutDate Imposta la data
PutTime Imposta l'ora
PutTimestamp Imposta la data,il giorno della settimana e l'ora
Timer Restituisce il numero di secondi trascorsi dalla mezzanotte in decimale

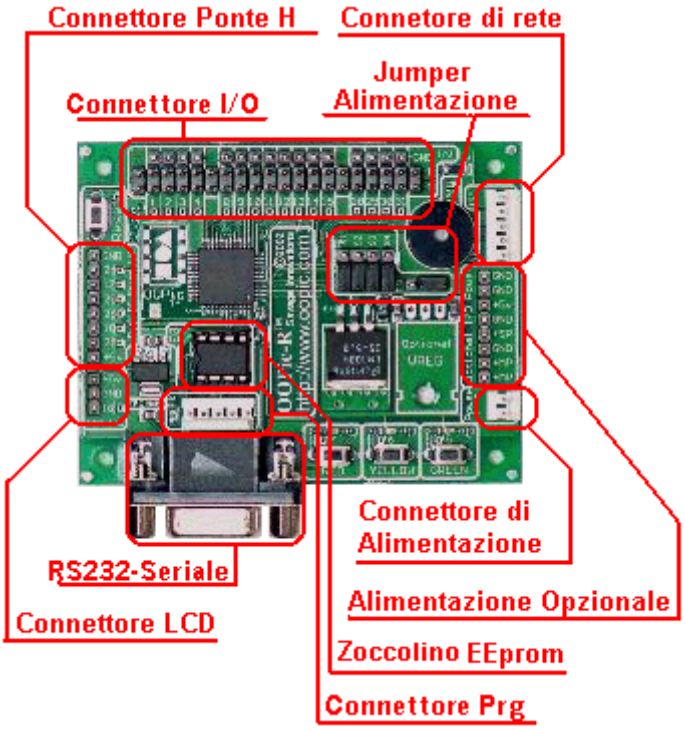
Funzioni per il controllo delle linee di I/O
ADCToCom1 Invia dati dall' ADC alla porta seriale
Com1ToDAC Invia dati dalla porta seriale all' ADC
CountTransitions Conta le transizioni logiche su un pin di input
DACPin Genera una tensione pseudo-analogica su un pin di uscita
FreqOut Genera delle sinusoidi duali su un pin di uscita
GetADC Restituisce una tensione analogica
GetPin Restituisce il livello logico di un pin di input
Capture Registra un treno di impulsi sul pin di input capture
OutputCapture Invia un treno di impulsi al pin di output capture
PlaySound Riproduce un suono utilizzando dati memorizzati nella EEPROM
PulseIn Misura la durata di un impulso su un pin di input
PulseOut Invia un impulso ad un pin di output
PutDAC Genera una tensione pseudo-analogica su un pin di uscita
PutPin Pone un pin in uno di quattro stati
RCTime Misura il tempo impiegato per la transizione su un pin

ShiftIn Fa scorrere i bit da un pin di I/O in una variabile byte
ShiftOut Fa scorrere i bit da una variabile byte su un pin di I/O

Gestione delle comunicazioni
Debug.Print Invia una stringa alla porta seriale Com1
DefineCom3 Definisce i parametri per una comunicazione seriale su un pin arbitrario
Get1Wire Riceve i bit di dato usando il protocollo Dallas 1-Wire
OpenCom Attiva una porta seriale RS-232
OpenSPI Attiva le comunicazioni SPI
Put1Wire Trasmette i dati usando il protocollo Dallas 1-Wire
SPICmd Comunicazioni SPI
X10Cmd Trasmette dati X-10

Vai a Robot “Pompiere”
Vai a Robot “Guardiano”
Vai a Possibili sviluppi
Torna all’indice

OOPic-R



La scheda OOPic-R include, oltre al microcontrollore OOPic, una porta seriale RS232, 16 linee I/O configurate per i servi RC (mentre le altre linee di I/O sono configurate per motori DC), LCD seriale, un connettore I2C, ecc. La scheda OOPIC-R è dotata di due (espandibili a 3) regolatori di tensione, tre pulsanti programmabili, tre indicatori led, uno speaker e del firmware OOPic2+. Il linguaggio di programmazione usato è l'OOpic Basic (completamente compatibile con Microsoft Visual Basic).

PROTOCOLLO DI CONTROLLO SERIALE (SCP)

Questa opzione introdotta nella versione firmware OOPic2+ permette di controllare l'OOPic via porta seriale. L'SCP permette a qualunque dispositivo connesso tramite la seriale di interagire con l'OOPic e fornisce delle opzioni di debug che consentono al programma residente nell'OOPic di essere fermato, di essere fatto avanzare passo per passo o di essere riavviato. L'intero set di istruzioni usate dall'SCP è composto da caratteri facilmente comprensibili dall'utente cosicchè l'OOPic possa essere controllato e programmato manualmente. Il baud rate di default è di 9600 BPS, ma può essere aumentato fino 38400 BPS o ridotto a 1200 BPS. Maggiori informazioni sono disponibili a questo link:

www.oopic.com/pgchap16.htm

Manuale software: http://www.oopic.com/PGFull.html#pgchap1.htm

ALIMENTAZIONE E REGOLATORI DI TENSIONE.

I tre regolatori di tensione dell'OOPic-R sono configurati in modo che le logiche della scheda abbiano a disposizione una corrente di 1A, mentre 5A sono disponibili per l' I/O. Un insieme di jumper di selezione permette ai vari blocchi di linee di I/O di essere connessi alle differenti alimentazioni. Oltre alle tensioni regolate viene fornito un ingresso di "Alimentazione servi" per collegare le linee di I/O ad una alimentazione esterna non regolata.

CONNETTORE I/O

Le linee di I/O della scheda OOPic-R sono divise in quattro gruppi, ciascuno dei quali è composto da Quattro porte di I/O; ogni porta ha tre pin: segnale, alimentazione e massa.
Connettore ponte H.
I connettore ponte H è composto da due linee PWM e da quattro linee di controllo digitali. Queste linee di I/O possono essere utilizzate per svolgere altre funzioni se non vengono usati dei motori DC.

CONNETTORE LCD

Il connettore LCD è formato da alimentazione, massa ed una linea di controllo. Questa linea di I/O può essere utilizzata per svolgere altre funzioni se non viene collegato un LCD.

PULSANTI E LED DI STATO

I tre pulsanti e LED di stato consentono di verificare lo stato delle linee di I/O.

CONNETTORE DI RETE

Il connettore di rete della scheda OOPic-R usa il protocollo I2C.

ALTOPARLANTE

L'altoparlante è connesso alla linea di I/O 21.

LED DI CORRETTA ALIMENTAZIONE

Il LED di corretta alimentazione mostra lo stato del clock della EEPROM. Se il LED è acceso significa che l'OOPic riceve abbastanza potenza da far funzionare il PIC il quale pilota l'ingresso di clock della EEPROM. Questo LED è indicato sulla scheda come EAC.

CONNETTORE PRG

Oltre a poter programmare l'OOPic tramite la porta seriale è possibile utilizzare il connettore Prg. Questa porta viene anche utilizzata da eventuali coprocessori che comunicano tramite il protocollo I2C; nel nostro caso questo connettore è collegato al sensore piroelettrico.

Pin	Nome	Dir	Descrizione	Funzione
1	LSDL		Local I2C Serial Data	Dati seriali I2C generati dal programmatore dell'OOPic durante la lettura/scrittura della EEPROM.
2	GND		Ground (massa)	0 Volt
3	LSCA		Local I2C Serial Clock	Clock seriale I2C generato dal programmatore dell'OOPic durante la lettura /scrittura della EEPROM.
4	+5		Alimentazione +5V	+5 Volt
5	RESET		Reset (Attivo basso)	Posto a livello basso per resettare l'OOPic durante la programmazione della EEPROM.

Vai a Robot "Pompiere"
Vai a Robot "Guardiano"
Vai a Possibili sviluppi
Torna all'indice

SERVI RC

Caratteristiche Tecniche HS 422:	
Supporti	boccole speciali
Coppia kg*cm	4.1kg (6Vdc)
Velocita' s/60°	0.16 (6Vdc)
Peso	45.5 g
Tipo Ingranaggi	Resina ad alta resistenza
Dimensioni	40 x 20 x 37 mm

Caratteristiche Tecniche HS 635HB:	
Supporti	boccole speciali
Coppia kg*cm	6kg (6Vdc)
Velocita' s/60°	0.15 (6Vdc)
Peso	44.5 g
Tipo Ingranaggi	Nylon
Dimensioni	40 x 20 x 38 mm

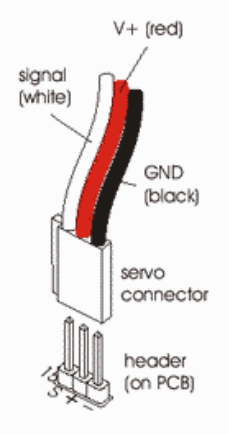
Un Servo RC tipicamente consiste in uno “scatolotto” di plastica di ridotte dimensioni da cui fuoriesce un perno in grado di ruotare in un angolo compreso tra 0 e 180 gradi e mantenere stabilmente la posizione raggiunta.

La rotazione del perno viene ottenuta tramite un motore in corrente continua ed un meccanismo di demoltiplica che consente di ottenere un'ottima coppia in fase di rotazione.

L'azionamento del motore viene effettuato tramite un circuito di controllo interno in grado di rilevare l'angolo di rotazione raggiunto dal perno tramite un potenziometro resistivo e bloccare il motore sul punto desiderato.
In dotazione ai servocomandi vengono anche fornite una o più squadrette forate da poter innestare sul perno per poter trasmettere il movimento ad altre parti meccaniche.
Sulla confezione c'è scritta la forza torcente (torque) a 4.8 e/o 6V (la forza si intende applicata a un centimetro di distanza dall'asse). Un servocomando da 3.5kg*cm genera una forza massima di 3.5kg alla distanza di 1 cm, 0.35kg alla distanza di 10cm ecc..

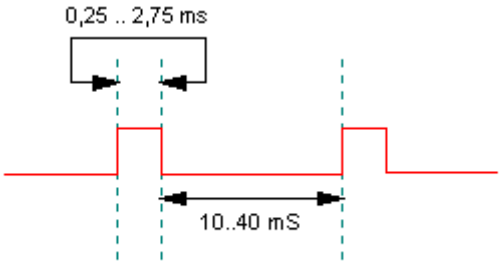
COME SI USA UN SERVO RC

I Servo RC sono nati per essere pilotati nel modo più semplice possibile. L'esigenza primaria era quella di poter effettuare la movimentazione senza l'ausilio di circuiterie troppo complesse o l'uso di sistemi a microprocessore. Un servo RC dispone solitamente di soli tre fili collegati ad un connettore femmina per pin strip a passo 2.54mm come visibile in figura:

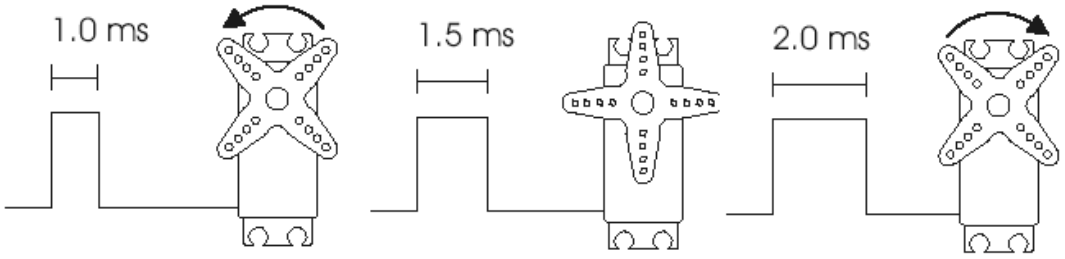


Due di questi fili sono riservati all'alimentazione in corrente continua a 5 volt (in realtà da 4.5 a 6.5). Il positivo è di colore rosso, il negativo di colore nero. Il terzo filo, normalmente di colore bianco o giallo, è riservato per il controllo del posizionamento. Su questo filo è necessario applicare un segnale impulsivo o PWM (dall'inglese Pulse Width Modulation) le cui caratteristiche sono "quasi" univoche per qualsiasi Servo RC disponibile in commercio. Per essere sicuri di riuscire a pilotare qualsiasi Servo RC il nostro circuito di pilotaggio dovrà essere in grado di inviare al servo circa 50 impulsi positivi al secondo (50Hz) di durata variabile in un intervallo massimo compreso tra 0.25ms e 2.75ms. Tra un impulso e l'altro non deve esserci una pausa inferiore a 10mS, altrimenti il circuito interno del servo può perdere la propria temporizzazione. Tra un impulso e l'altro possono passare anche diverse decine di mS, però più gli impulsi sono distanti tra loro e meno forza il servo riesce ad applicare (lo si sente vibrare alla frequenza degli impulsi e il controllo diventa lento, e poco preciso).

IMPULSI DI COMANDO DI UN SERVO

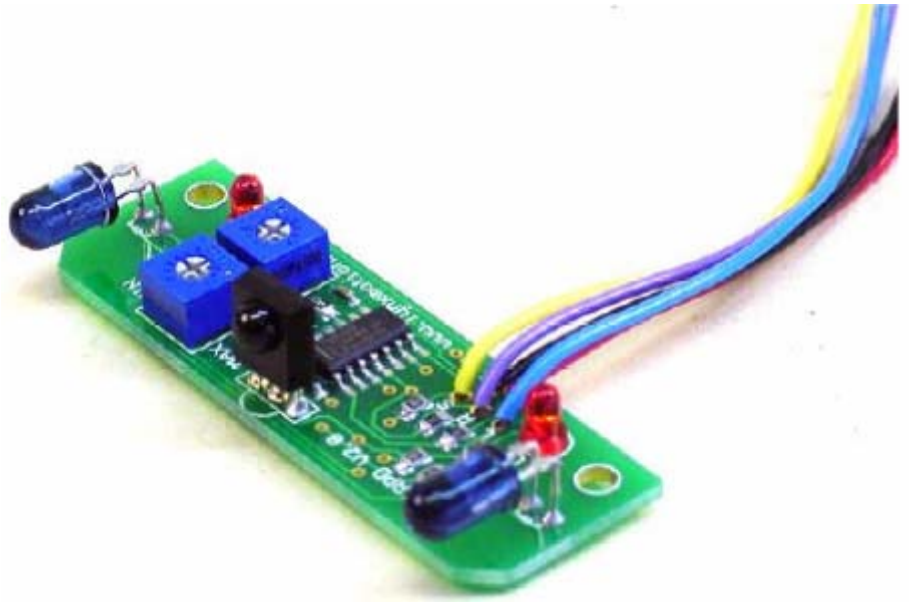


Generalmente con un impulso di durata pari a 1.5ms il perno del servo RC si posiziona esattamente al centro del suo intervallo di rotazione. Da questo punto il perno può ruotare fino a -90 gradi (senso antiorario) se l'impulso fornito ha una durata inferiore a 1.5ms e fino +90 gradi (senso orario) se l'impulso fornito ha durata superiore a 1.5ms. Il rapporto esatto tra la rotazione del perno e la larghezza dell'impulso fornito può variare tra i vari modelli di servo. Esempio di relazione tra il segnale PWM e la rotazione del perno:



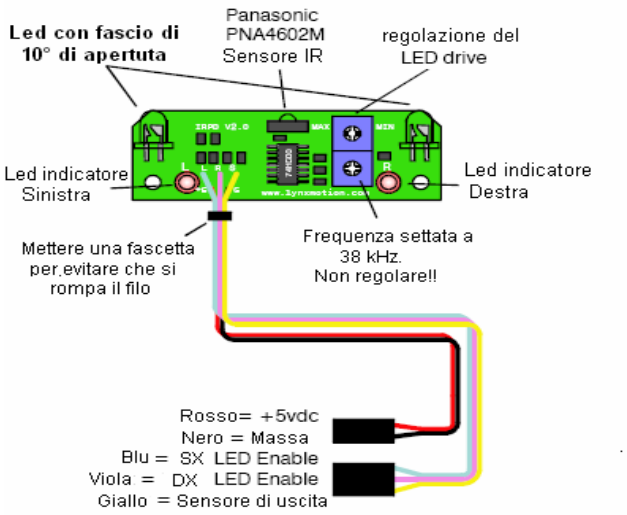
Vai a Robot “Pompiere”
Vai a Robot “Guardiano”
Vai a Possibili sviluppi
Torna all’indice

SENSORE DI PROSSIMITA' AD INFRAROSSI (IRPD-V7)

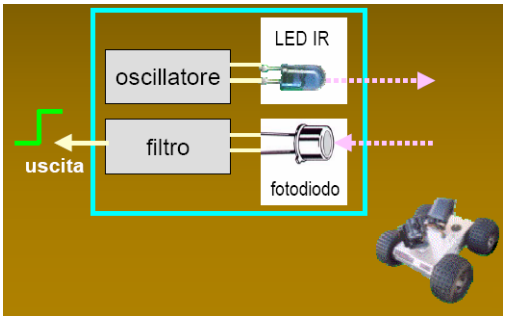


SPECIFICHE

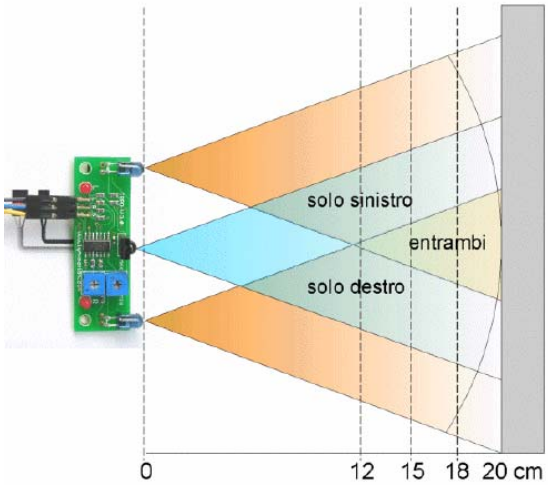
- Tipo di sensore = Reflective IR
- Sensore IR = Panasonic PNA4602M
- Tipo di LED IR = Ampiezza fascio 10º
- I/O = Tre linee digitali, 2 output, 1 input
- Raggio minimo = Circa 10,16cm
- Raggio massimo= Circa 66cm
- Alimentazione = 5V DC stabilizzata
- Assorbimento corrente = 8mA
- Dimensioni PCB = 5,84 x .1,9cm


Led con fascio di 10° di apertuta
Panasonic PNA4602M Sensore IR
regolazione del LED drive
Led indicatore Sinistra
Led indicatore Destra
Mettere una fascetta per.evitare che si rompa il filo
Frequenza settata a 38 kHz. Non regolare!!
Rosso= +5vdc
Nero = Massa
Blu = SX LED Enable
Viola = DX LED Enable
Giallo = Sensore di uscita

L'IRPD-V7 è un modulo con microcontrollore che gestisce un sensore infrarosso in grado di rilevare ostacoli con un angolo di circa 180°. Esso può individuare degli oggetti in tre diversi quadranti e la distanza minima di rilevamento è regolabile fino a pochi centimetri: per far ciò usa un sensore Panasonic PNA4602M in coppia con due LED IR. Il modulo Panasonic contiene amplificatori integrati, filtri (eliminano il rumore dovuto alla luce solare) e un limitatore. I LED sono modulati con un oscillatore regolabile e variando la loro corrente di pilotaggio si può controllare la sensibilità del sensore.

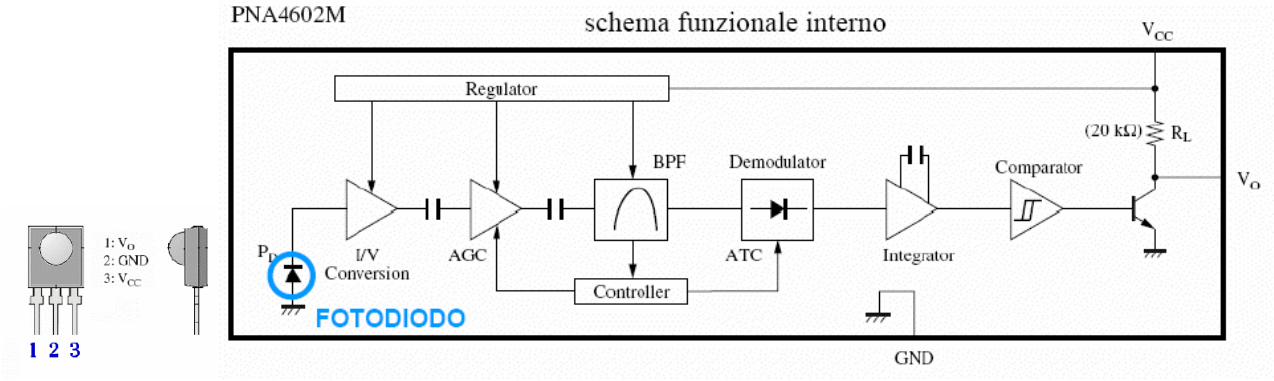


Il microcontrollore abilita alternativamente i LED e verifica se vi sono riflessioni indicanti la presenza di un oggetto sulla sinistra,sulla destra o al centro. L'assorbimento di corrente è basso, circa 8mA, dato che i LED non sono sempre accesi. Le tre linee digitali servono per interfacciare l'IRPD con un microcontrollore esterno.

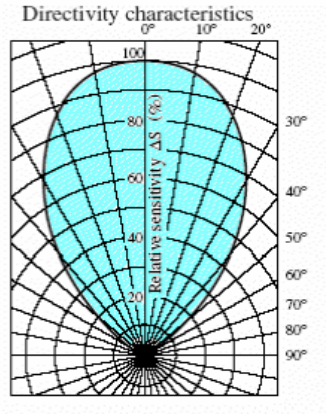


PNA 4602M

Di seguito vi è lo schema a blocchi del ricevitore ad infrarossi. Grazie al fotodiodo è in grado di ricevere le riflessioni del segnale dei led. E' sensibile ai soli raggi infrarossi in quanto ha un involucro in resina scura.



L'RX ha una lente integrata che garantisce una buona direzionalità.



MONTAGGIO

L'altezza ideale a cui montare il sensore è di circa 10-15cm dal suolo: se venisse montato più in basso potrebbe essere necessario puntare i LED più in alto. Esso deve essere sistemato in posizione orizzontale assicurandosi che nessuna parte mobile del robot passi all'interno del campo visivo causando degli errori nella rilevazione.

VERIFICA FUNZIONAMENTO

1) Applicare +5vdc e massa ai fili rosso e nero.

2) Applicare +5vdc all'enable SX (filo blu).

3) Posizionare la vite del Led drive in posizione intermedia.

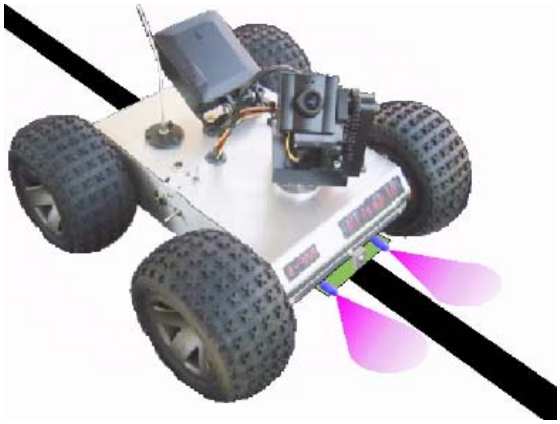
4) Montare l' IRPD in modo che non punti verso alcun oggetto. I LED dovrebbero essere spenti.

5) Mettere la mano davanti al sensore spostata di circa 10cm sulla sinistra rispetto al centro. Dovrebbe accendersi solo il LED sinistro: se non è così togliere l'alimentazione e controllare I collegamenti.

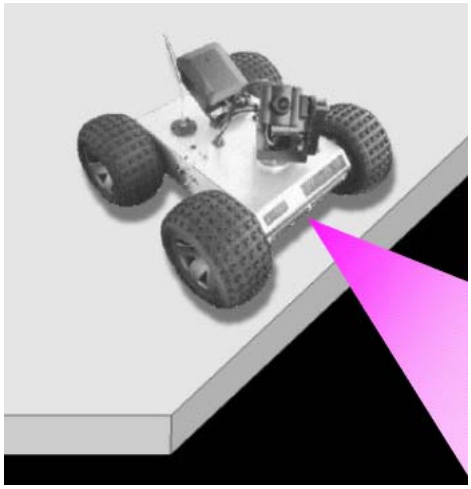
6) Ora allontanare lentamente la mano dal sensore e prendere nota della distanza a cui il LED comincia a lampeggiare: questa indica il range massimo in cui può essere individuato un ostacolo. Se la distanza risulta essere troppo ampia, regolare il led DRIVE girando la vite

verso destra per diminuire la corrente di pilotaggio dei LED IR e di conseguenza anche il range MAX. Viceversa se la distanza è troppo piccola girare la vite verso sinistra.
7) Togliere l'alimentazione dal LED enable SX e applicarla al LED enable DX (filo viola). Ripetere i passi 5 e 6 per il lato destro.

ALTRI POSSIBILI UTILIZZI



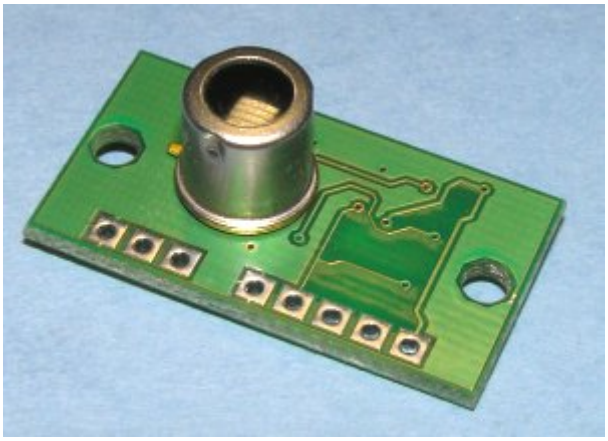
un sensore di prossimità IR può essere utilizzato per seguire una traccia



Per evitare un bordo invece si potrebbe utilizzare un solo led
Per maggiori informazioni: http://www.lynxmotion.com/images/data/irpd-v7.pdf

Vai a Robot "Pompiere"
Vai a Robot "Guardiano"
Vai a Possibili sviluppi
Torna all'indice

SENSORE PIROELETTRICO TPA 81



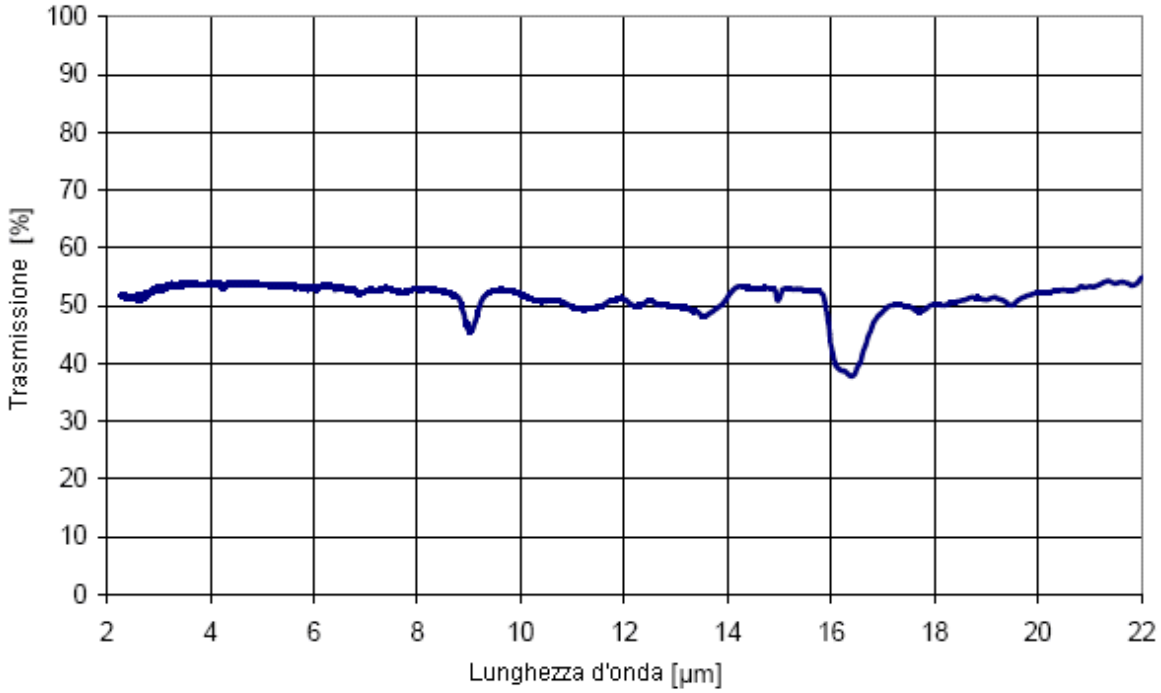
Caratteristiche Tecniche:	
Tensione Operativa	5V
Corrente Operativa Tipica	mA
Accuratezza Range 4-10°C	+/-3°C
Accuratezza Range 10-100°C	+/-2°C +/-2%
Campo di rilevamento (FOV)	41° x 6° (8 pixel of 5° x 6°)
Uscite	1 Temp ambiente + 8 Temp degli 8 pixel
Comunicazione	Protocollo I2C
Controllo Servo	32 passi per 180° di rotazione
Dimensioni	31 x 18 mm

Il sensore TPA81 e' costituito da una matrice di sensori sensibili alla gamma infrarossa in un range che va dai 2um ai 22um. La banda e' simile a quella rilevata dai sensori PIR che si trovano negli allarmi / antifurto o nei dispositivi che accendono le luci. La caratteristica di questi sensori, e' che possono rilevare soltanto il movimento degli oggetti, e questo li rende limitati nel settore della robotica. Ad esempio, non possono essere utilizzati per misurare la temperatura di una sorgente calda statica.

Un tipo differente di sensori sono i 'thermopile array'. Questi vengono utilizzati nei misuratori di temperatura all'infrarosso, che non richiedono il contatto con la sorgente da misurare. Hanno un ampio campo di rilevamento di circa 100° e spesso sono utilizzati con una lente che restringe il campo di misura a circa 12°, molto piu' utile in un'applicazione di misura.
Recentemente sono disponibili sul mercato dei sensori dotati di lenti al silicone, questo e' il tipo utilizzato nel TPA81.
Il TPA81 può misurare la temperatura di 8 punti adiacenti, contemporaneamente. Puo' anche controllare un servo per muovere, sull'asse orizzontale, il sensore e costruire una immagine dei valori termici rilevati. Il TPA81 riesce ad individuare la fiamma di una candela a 2 metri di distanza ed e' insensibile alla luce ambientale.

SPETTRO DI RISPOSTA

In figura vi è rappresentata ad ogni lunghezza d'onda la percentuale di trasmissione:

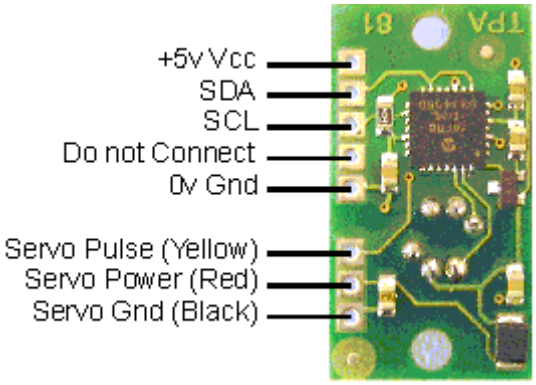


CAMPO VISIVO (Field Of View)

Tipicamente il campo visivo del TPA81 è 41° in altezza e 6° in larghezza esso è formato da 8 pixel di 5.12°x 6°.

CONNESSIONI

Tutte le comunicazioni con il TPA81 avvengono via bus I2C (connettore standard a 5 pin). Se non si ha familiarità con l'I2C è disponibile un **tutorial**. Le linee SCL e SDA devono essere collegate a +5V tramite dei resistori di pull-up(1,8kΩ). Alcuni circuiti come l' OOPic hanno già i resistori di pull-up e quindi non è necessario aggiungerne altri. Alla "porta servo" può essere connesso direttamente un servo RC standard. Tramite un microcontrollore, si possono inviare i comandi per posizionare il servo al TPA81 il quale a sua volta genera gli impulsi per pilotare lo stesso.



REGISTRI

Il TPA81 viene visto dal microcontrollore come un insieme di 10 registri.

Registro	Lettura	Scrittura
0	Versione sofware	Registro comandi
1	Temperatura ambiente °C	Usato per la calibrazione – non scrivere
2	Temperatura Pixel 1 °C	Usato per la calibrazione – non scrivere
3	Pixel 2	Usato per la calibrazione – non scrivere
4	Pixel 3	N/A
5	Pixel 4	N/A
6	Pixel 5	N/A
7	Pixel 6	N/A
8	Pixel 7	N/A
9	Pixel 8	N/A

Solo i registri 0,1,2 e 3 possono essere scritti. Il registro 0 è il "registro comandi", usato per impostare la posizione del servo e quando viene cambiato l'indirizzo I2C del TPA81; leggendo da esso si ottiene la versione del software. I registri 1,2 e 3 sono usati per la calibrazione del sensore e non vanno modificati per non cancellare i dati della stessa.
Ci sono 9 valori di temperature disponibili in lettura, tutti in gradi centigradi(°C).Nel registro 1 è contenuta la temperatura ambiente, mentre i registri da 2 a 9 contengono le temperature corrispondenti agli 8 pixel. Ogni 40mS sono disponibili le nuove rilevazioni di temperatura all'interno dei registri.

POSIZIONE SERVO

I comandi da 0 a 31 impostano la posizione del servo. Ci sono 32 passi(0-31) che rappresentano la rotazione di 180°. Il calcolo da effettuare per ottenere la durata dell'impulso necessaria per spostare il servo in una determinata posizione è: SERVO_POS*60+540uS. Quindi la durata dell'impulso varia da 0,54mS a 2,4mS. Con passi di 60us. Un valore non compreso tra 0 e 31 non fa muovere il servo.

Comando		Azione
Decimale	Hex	
0	0x00	Posizione di inizio corsa
1-30	0x01-0x1E	Posizioni intermedie
31	0x1F	Posizione di fine corsa
160	0xA0	1° comando sequenza cambiamento indirizzo I2C
165	0xA5	3° comando sequenza cambiamento indirizzo I2C
170	0xAA	2° comando sequenza cambiamento indirizzo I2C

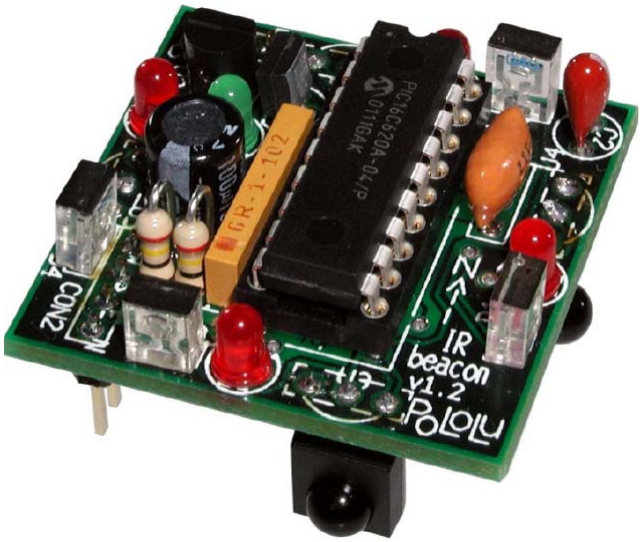
CAMBIARE L' INDIRIZZO BUS I2C

Per cambiare l'indirizzo I2C del TPA81 bisogna avere solo un circito connesso al bus. Scrivere la sequenza di 3 comandi nell'ordine esatto seguiti dall'indirizzo. Ad esempio, per cambiare l'indirizzo del TPA81 da 0xD0 (ind. di default di fabbrica) a 0xD2, inviare la seguente sequenza all'indirizzo 0xD0; (0xA0, 0xAA, 0xA5, 0xD2). La sequenza deve essere inviata al "registro comandi" alla locazione 0, il che implica 4 cicli di scrittura sul bus I2C; inoltre ci deve essere un ritardo di almeno 50mS tra ogni byte delle sequenza di

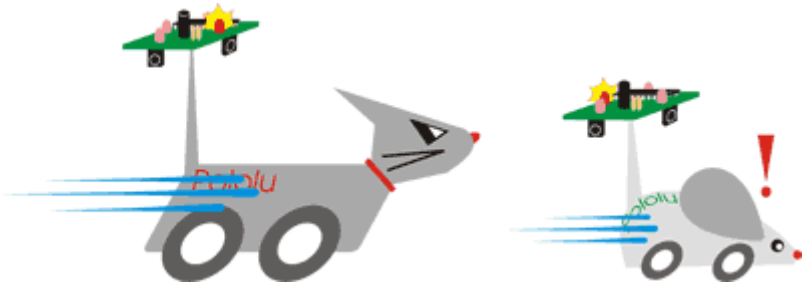
cambiamento dell'indirizzo. Una volta terminato annotare il nuovo indirizzo del sensore, dato che, nel caso venisse smarrito, potrebbe essere recuperate solo provando un indirizzo alla volta e verificandone la risposta. Il TPA81 può essere associate a uno qualunque di questi otto indirizzi: 0xD0, 0xD2, 0xD4, 0xD6, 0xD8, 0xDA, 0xDC, 0xDE.

Vai a Robot "Pompiere"
Vai a Robot "Guardiano"
Vai a Possibili sviluppi
Torna all'indice

SEGNALATORE DI POSIZIONE INFRAROSSI (BEACON)



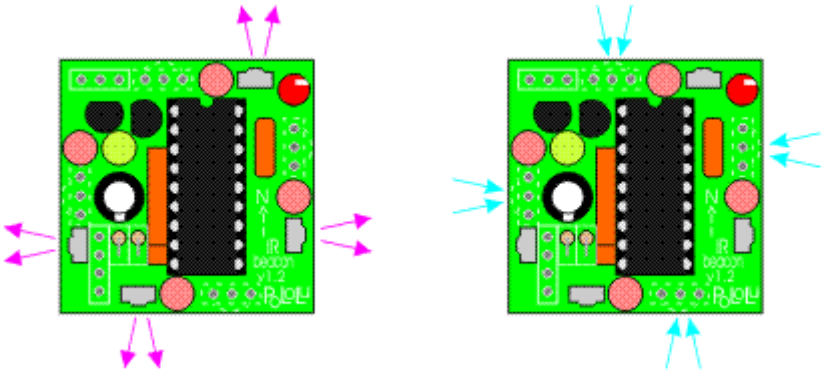
Il segnalatore di posizione a infrarossi è un piccolo dispositivo che permette ad una coppia di robot di localizzarsi a vicenda ed interagire o ad un singolo robot di raggiungere un punto definito.



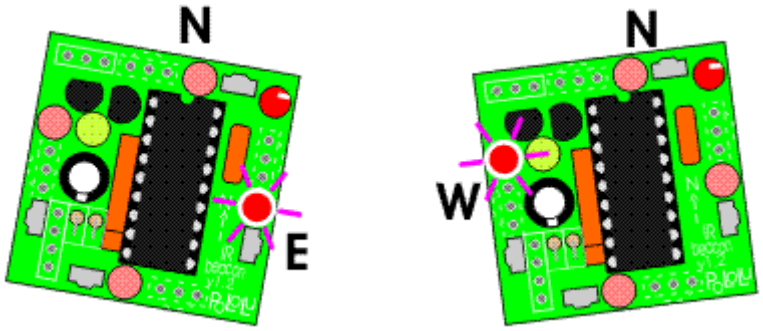
Caratteristiche	
Dimensioni PCB:	3,3cm x 3,3cm
Frequenza di modulazione IR:	56 kHz
Frequenza di refresh dell'output	20 Hz
Range di rilevameto:	15,2 cm a 6,1 m
Tensione di alimentazione: Consumo di c orrente media	5.1-10 V 50 mA
Livello tensione dati:	0 e 5 V
Numero di sensori IR :	4
Componenti nel kit:	25

COME FUNZIONA IL BEACON

Il beacon funziona trasmettendo e ricevendo segnali IR. Ogni beacon ha quattro emettitori e quattro ricevitori infrarosso. I beacon alternano la trasmissione e la ricezione in modo da individuare senza commettere errori il segnale proveniente da un altro emettitore.



Il ciclo di trasmissione e ricezione viene eseguito più di mille volte al secondo e un piccolo microcontrollore monitora i quattro ricevitori e stabilisce la direzione in cui si trova un altro beacon indicandola con l'accensione di uno dei quattro led rossi.

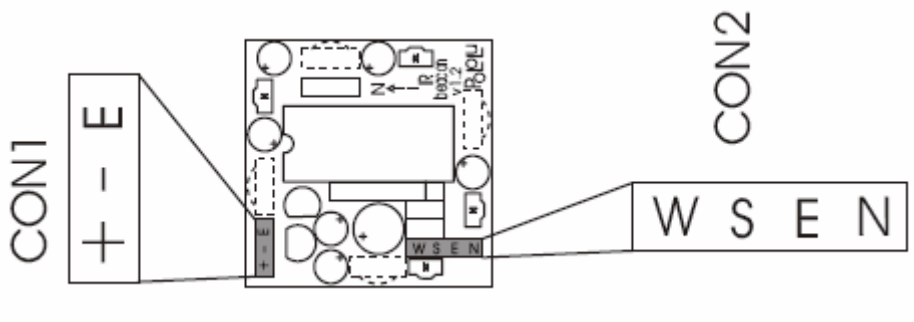


L'interfacciamento del beacon è abbastanza semplice, ha quattro uscite digitali (ognuna corrispondente ad un lato)che vengono portate a livello basso quando il sensore del corrispondente lato individua l'altro beacon. Si può stabilire la direzione in cui si trova un altro beacon, con una tolleranza di pochi gradi, ruotando il beacon in un verso e poi nell'altro annotando il punto in cui cambia il lato attivo. Un input di enable permette di scegliere la modalità -low power mode (Standby)- o -active mode-(in funzione)

MONTAGGIO E CONNESSIONI SUL ROBOT

Il beacon deve essere montato nel punto più alto del robot in modo che nessuna parte dello stesso blocchi la luce IR trasmessa e ricevuta.

Per istruzione più dettagliate sul montaggio dei componenti consultare http://www.pololu.com/products/pololu/0001/beacon_guide.pdf



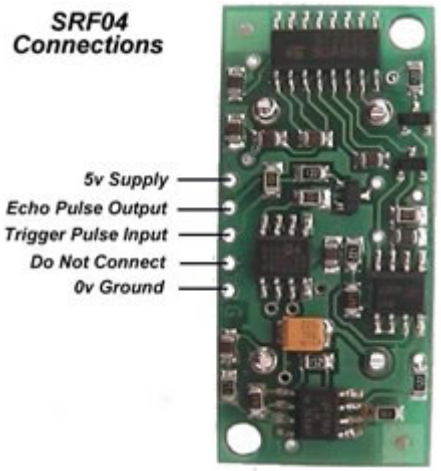
Sul beacon ci sono due connettori: CON1, a cui vanno collegati gli ingressi di alimentazione e di abilitazione,CON2 invece fornisce le uscite.
Il pin di enable è opzionale ed è posto a livello alto per default. Per disabilitare la trasmissione portare il pin di enable a livello basso: a questo punto il led verde lampeggia più lentamente ad indicare che il circuito è in standby.

TEST

Dopo aver assemblato il beacon connetterlo all'alimentazione facendo attenzione alla polarità e lasciando il pin di abilitazione disconnesso o impostato a livello alto. Se è alimentato correttamente i LED rossi dovrebbero accendersi e poi spegnersi nel seguente ordine N,E,S,O: a questo punto il LED verde inizia a lampeggiare indicando che il beacon ha cominciato a trasmettere. Se si ha un altro beacon accenderlo e verificare che su ciascuno di essi si accenda il LED rosso corrispondente alla direzione in cui si trova l'altro beacon. Un ultimo controllo effettuabile è quello di verificare che il beacon smetta di trasmettere quando la linea di abilitazione viene posta a livello basso (il LED verde lampeggia più lentamente e le uscite passano a livello alto facendo spegnere tutti i LED rossi).Se si riscontrano dei problemi durante questa procedura controllare che l'assemblaggio sia corretto ed eventualmente consultare la sezione relativa alla risoluzione dei problemi sul manuale utente (http://www.pololu.com/products/pololu/0001/beacon_guide.pdf).

Vai a Robot "Pompiere"
Vai a Robot "Guardiano"
Vai a Possibili sviluppi
Torna all'indice

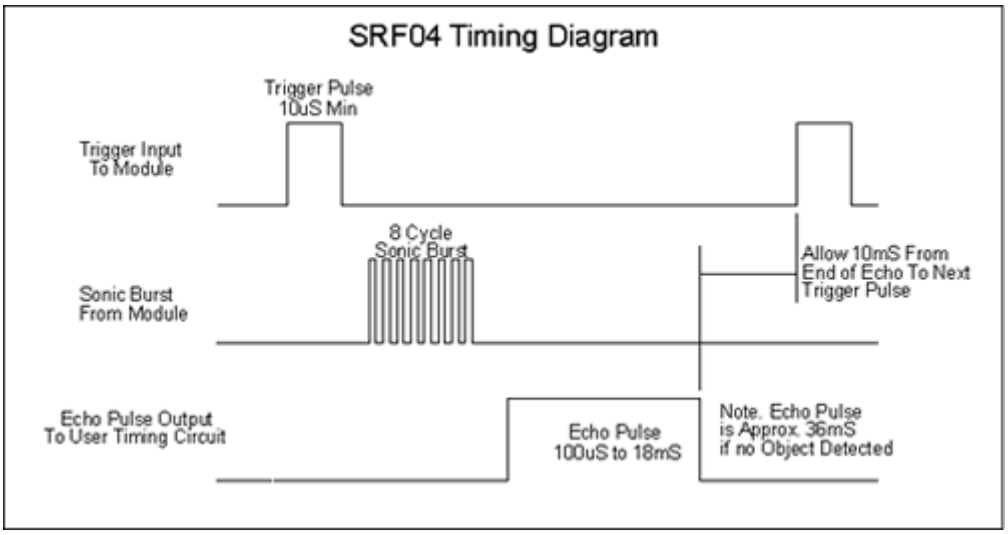
SENSORE DI DISTANZA AD ULTRASUONI – SRF04



L'**SRF04** e' un sensore ad ultrasuoni che unisce delle ottime prestazioni ad un costo conveniente: esso è composto da un emettitore e da un ricevitore di ultrasuoni Questo sensore e' dotato di un microcontrollore che assolve tutte le funzioni di calcolo ed elaborazione, e' sufficiente inviare un impulso e leggere l'eco di ritorno per stabilire la distanza dell'ostacolo o dell'oggetto che si trova davanti. Riesce ad individuare ostacoli ed oggetti, anche di piccole dimensioni, da 3cm fino a 3mt. Come tutti i sensori di questo tipo, e' insensibile alla luce ambientale, quindi e' ottimo per essere usato all'esterno. Da notare che il campo visivo di questo sensore e' abbastanza ampio, sebbene questo si possa ottimizzare regolando la soglia, in determinate applicazioni potrebbe essere preferibile un sensore con un campo piu' ristretto.

Caratteristiche Tecniche:	
Tensione Operativa	5V
Corrente Operativa Tipica	30mA - 50mA Max
Frequenza	40 Khz
Portata	3cm - 3mt
Sensibilita'	Rileva un oggetto da 3cm di diametro a 2mt
Trigger di input	10uS Min. Impulso di livello TTL
Echo Pulse	Livello disegnale TTL positivo, larghezza proporzionale alla distanza dell'oggetto
Dimensioni	43 x 20 x H 17 mm

Il diagramma temporale del SFR04 è mostrato nella figura successiva.



Per iniziare ad effettuare la misura di distanza è necessario fornire un impulso di 10uS all'ingresso di trigger. L'SRF04 emette quindi un treno di 8 impulsi ad ultrasuoni alla frequenza di 40khz e pone la sua linea di eco a livello alto. Il sensore si predispone poi per ascoltare l'eco e appena ne rileva una pone la linea di eco a livello basso. La linea di eco è quindi un impulso la cui larghezza è proporzionale alla distanza del sensore dall'oggetto. Misurando la durata dell'impulso è possibile calcolare la distanza in pollici/centimetri o in qualunque altra unità di misura. Se non viene rilevato nulla l' SRF04 pone come la linea di eco a livello basso dopo circa 36mS.

Il circuito utilizza un PIC12C508, che esegue le funzioni di controllo, e trasduttori piezoelettrici standard. Il trasduttore trasmittente può essere pilotato direttamente dal PIC. La tensione di pilotaggio a 5v può essere utilizzata per rilevare oggetti di grosse dimensioni, mentre per oggetti più piccoli è necessaria una tensione più elevata (MAX 20V) fornita dal circuito integrato MAX232.

Il ricevitore è un circuito a due stadi con amplificatori operazionali. Ogni stadio ha un guadagno pari a 24 per un totale di 576. Questo valore è vicino a quello massimo (25) ottenibile con l'LMC6032, che ha un prodotto banda-guadagno di 1MHz. L'uscita dell' amplificatore è inviata ad un comparatore LP311 . Una parte di segnale viene retroazionata positivamente fornendo un anello di controllo per dare un output stabile.

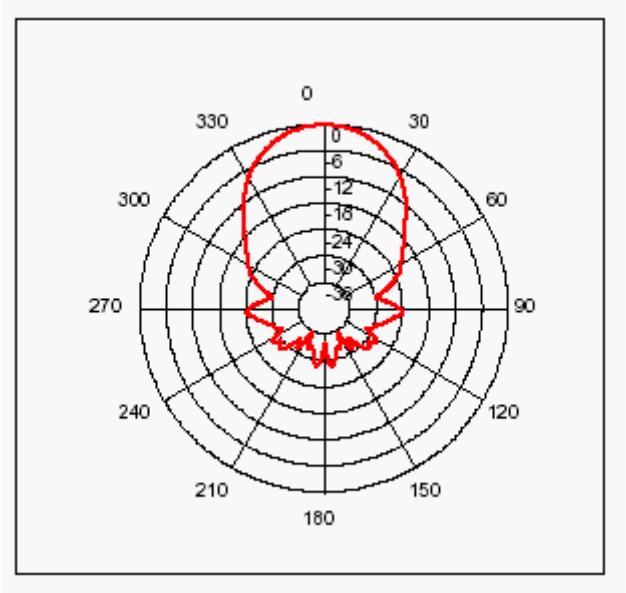
Misurare distanze più piccole di tre centimetri è problematico perché il ricevitore rischia di rilevare, invece dell'eco di ritorno, il treno di impulsi emesso dal trasmettitore; inoltre i

trasduttori piezoelettrici continuano a risuonare per un po' di tempo (fino a 1mS) dopo che la tensione di pilotaggio è stata rimossa.

CALCOLO DELLA DISTANZA

L'SRF04 fornisce un impulso di eco proporzionale alla distanza. Se l'ampiezza dell'impulso è misurata in uS, allora dividendo per 58 si ottiene la distanza in cm, mentre dividendo per 148 si ottiene la distanza in pollici.

Di seguito è riportato il diagramma di irradiazione del sensore.



Manuale in inglese: http://www.robot-electronics.co.uk/htm/srf04tech.htm

Vai a Robot "Pompiere"
Vai a Robot "Guardiano"
Vai a Possibili sviluppi
Torna all'indice

VENTOLA

La ventola ha un diametro di 50 mm ed è completa di motore, condensatore ceramico gia' montato, aletta di raffreddamento, cavetto e spina tipo servo R/C.

CARATTERISTICHE TECNICHE

Volts(V)	6 2-7.2
Ampere(A)	2.3-3
Spinta(g)	48-73
Potenza(W)	13.8-21.6
Efficienza (g/W)	3.478-3.379

CIRCUITO DI PILOTAGGIO

Per pilotare la ventola abbiamo dovuto progettare e costruire un apposito controllore. I motivi per cui ciò si è reso necessario sono l'alta corrente assorbita dal motore (non era sufficiente quella fornita dalle schede dei microcontrollori) e la tensione fornita dalle batterie che era troppo elevata ed avrebbe danneggiato la ventola.

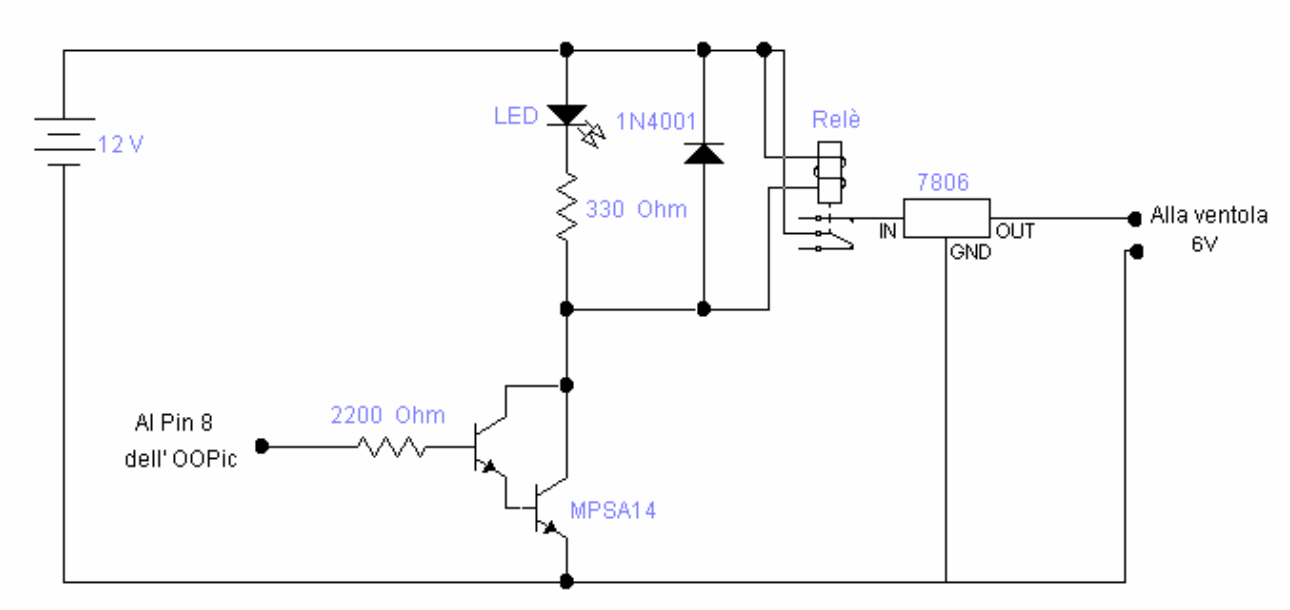
L'alimentazione del circuito realizzato è prelevata direttamente dalla batteria (12V) e tramite un regolatore di tensione integrato (7806) viene ridotta a 6V.

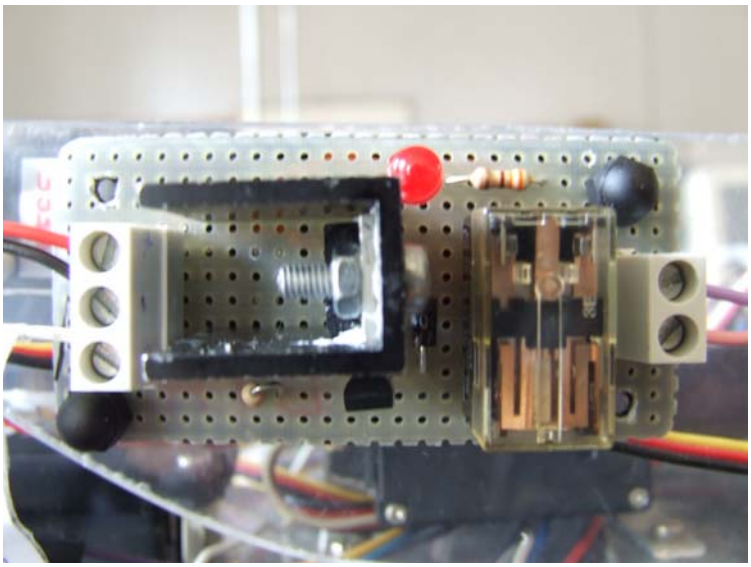
Il pilotaggio della ventola avviene per mezzo di un segnale di controllo, proveniente dalla porta 8 dell'OOPic , il quale viene portato a livello logico alto (5 V) per azionare la ventola quando il sensore piroelettrico rileva una fiamma ed il braccio è già stato sollevato. Una volta spenta la fiamma il segnale viene riportato a livello basso e la ventola si spegne.

L'ingresso del segnale di controllo è collegato alla base di un transistor di tipo Darlington (MPSA14) per essere sicuri che la corrente erogata dallo stesso sia sufficiente ad eccitare la

bobina del relè che connette l'alimentazione all'ingresso del regolatore la cui uscita è collegata alla ventola. Gli altri componenti del circuito sono: un diodo LED con relativa resistenza da 330Ω per limitare la corrente; una resistenza da 2,2KΩ collegata alla base del transistor per limitare la corrente richiesta al microcontrollore; un diodo di protezione (1N4001) collegato in parallelo alla bobina del relè per evitare picchi di correnti.

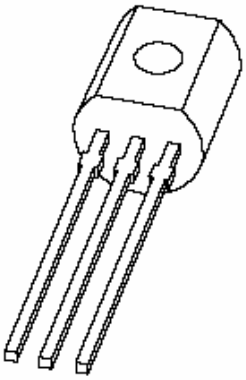
SCHEMA ELETTRICO





Di seguito sono riportati i fogli tecnici del transistor MPSA14, del diodo 1N4001 e del regolatore di tensione 7806.

DATA SHEET



MPSA14
NPN Darlington transistor

Product specification
Supersedes data of 2003 Oct 22

2004 Aug 20

Philips
Semiconductors

PHILIPS

NPN Darlington transistor MPSA14

FEATURES

- Low current (max. 100 mA)
- High voltage (max. 300 V).

APPLICATIONS

- General purpose switching and amplification.

DESCRIPTION

PNP high-voltage transistor in a TO-92; SOT54 plastic package. NPN complement: MPSA42.

PINNING

PIN	DESCRIPTION
1	collector
2	base
3	emitter

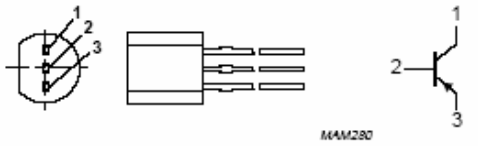


Fig.1 Simplified outline (TO-92; SOT54) and symbol.

LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 60134).

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V_{CBO}	collector-base voltage	open emitter	–	–300	V
V_{CEO}	collector-emitter voltage	open base	–	–300	V
V_{EBO}	emitter-base voltage	open collector	–	–5	V
I_C	collector current (DC)		–	–100	mA
I_{CM}	peak collector current		–	–200	mA
I_{BM}	peak base current		–	–100	mA
P_{tot}	total power dissipation	$T_{amb} \leq 25$ °C	–	625	mW
T_{stg}	storage temperature		–65	+150	°C
T_j	junction temperature		–	150	°C
T_{amb}	operating ambient temperature		–65	+150	°C

THERMAL CHARACTERISTICS

SYMBOL	PARAMETER	CONDITIONS	VALUE	UNIT
$R_{th\ j\text{-}a}$	thermal resistance from junction to ambient	note 1	200	K/W

Note

1. Transistor mounted on an FR4 printed-circuit board.

CHARACTERISTICS

T_j = 25 °C unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
I_{CBO}	collector cut-off current	I_E = 0; V_{CB} = –200 V	–	–250	nA
I_{EBO}	emitter cut-off current	I_C = 0; V_{BE} = –3 V	–	–100	nA
h_{FE}	DC current gain	V_{CE} = –10 V; note 1			
		I_C = –1 mA	25	–	
		I_C = –10 mA	40	–	
		I_C = –30 mA	25	–	
V_{CEsat}	collector-emitter saturation voltage	I_C = –20 mA; I_B = –2 mA; note 1	–	–500	mV
V_{BEsat}	base-emitter saturation voltage	I_C = –20 mA; I_B = –2 mA; note 1	–	–900	mV
C_c	collector capacitance	$I_E = i_e$ = 0; V_{CB} = –20 V; f = 1 MHz	–	6	pF
f_T	transition frequency	I_C = –10 mA; V_{CE} = –20 V; f = 100 MHz	50	–	MHz

Note

1. Pulse test: $t_p \leq 300\ \mu s$; $\delta \leq 0.02$.

FAIRCHILD
SEMICONDUCTOR®

1N4001 - 1N4007

Features

- Low forward voltage drop.
- High surge current capability.

DO-41
COLOR BAND DENOTES CATHODE

General Purpose Rectifiers

Absolute Maximum Ratings*

T_A = 25°C unless otherwise noted

Symbol	Parameter	Value							Units
		4001	4002	4003	4004	4005	4006	4007	
V_{RRM}	Peak Repetitive Reverse Voltage	50	100	200	400	600	800	1000	V
$I_{F(AV)}$	Average Rectified Forward Current, .375 " lead length @ T_A = 75°C	1.0							A
I_{FSM}	Non-repetitive Peak Forward Surge Current 8.3 ms Single Half-Sine-Wave	30							A
T_{stg}	Storage Temperature Range	-55 to +175							°C
T_J	Operating Junction Temperature	-55 to +175							°C

*These ratings are limiting values above which the serviceability of any semiconductor device may be impaired.

Thermal Characteristics

Symbol	Parameter	Value	Units
P_D	Power Dissipation	3.0	W
$R_{\theta JA}$	Thermal Resistance, Junction to Ambient	50	°C/W

Electrical Characteristics

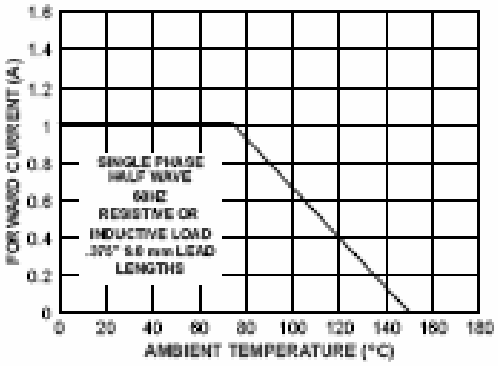
T_A = 25°C unless otherwise noted

Symbol	Parameter	Device							Units
		4001	4002	4003	4004	4005	4006	4007	
V_F	Forward Voltage @ 1.0 A	1.1							V
I_R	Maximum Full Load Reverse Current, Full Cycle T_A = 75°C	30							µA
I_R	Reverse Current @ rated V_R T_A = 25°C T_A = 100°C	5.0 500							µA µA
C_T	Total Capacitance V_R = 4.0 V, f = 1.0 MHz	15							pF

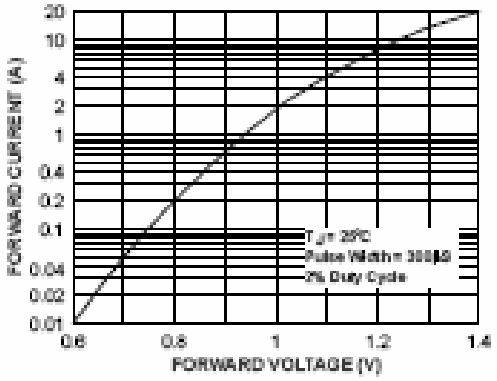
©2000 Fairchild Semiconductor Corporation

Typical Characteristics

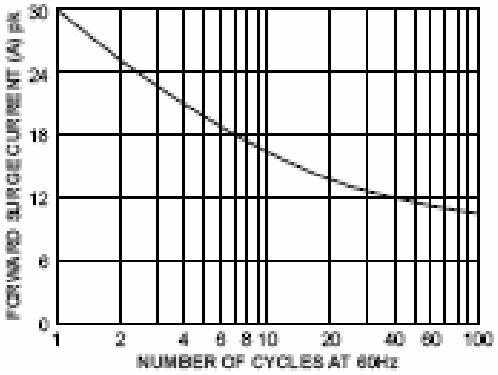
Forward Current Derating Curve



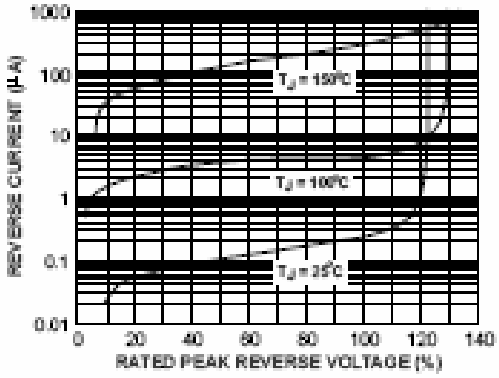
Forward Characteristics



Non-Repetitive Surge Current



Reverse Characteristics



©2003 Fairchild Semiconductor Corporation

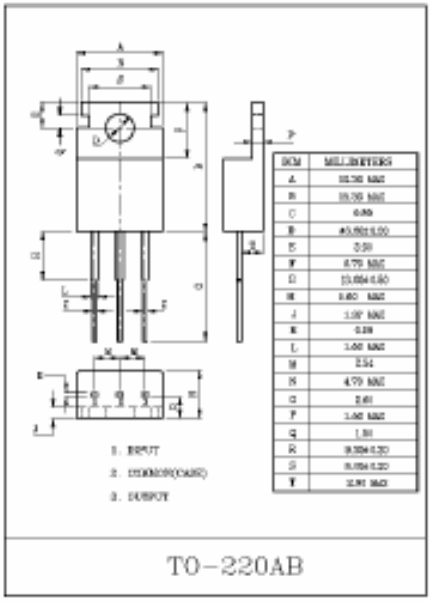
THREE TERMINAL POSITIVE VOLTAGE REGULATORS
5V, 6V, 8V, 9V, 10V, 12V, 15V, 18V, 20V, 24V.

FEATURES

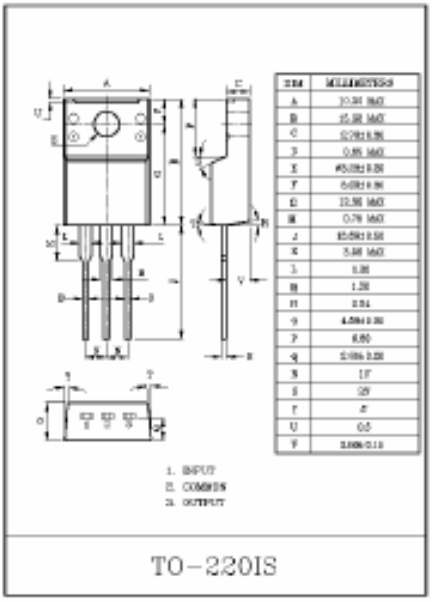
- Suitable for C-MOS, TTL, the Other Digital IC's Power Supply.
- Internal Thermal Overload Protection.
- Internal Short Circuit Current Limiting.
- Output Current in Excess of 1A.
- Satisfies IEC-65 Specification. (International Electronical Commission).

MAXIMUM RATINGS (Ta=25℃)

CHARACTERISTIC		SYMBOL	RATING	UNIT
Input Voltage	KIA7805AP/API~ KIA7815AP/API	V_{IN}	35	V
	KIA7818AP/API~ KIA7824AP/API		40	
Power Dissipation (Tc=25℃)		P_D	20.8	W
Power Dissipation (Without Heatsink)	KIA7805API~ KIA7824API	P_D	2.0	W
Operating Junction Temperature		T_j	-30~150	℃
Storage Temperature		T_{stg}	-55~150	℃

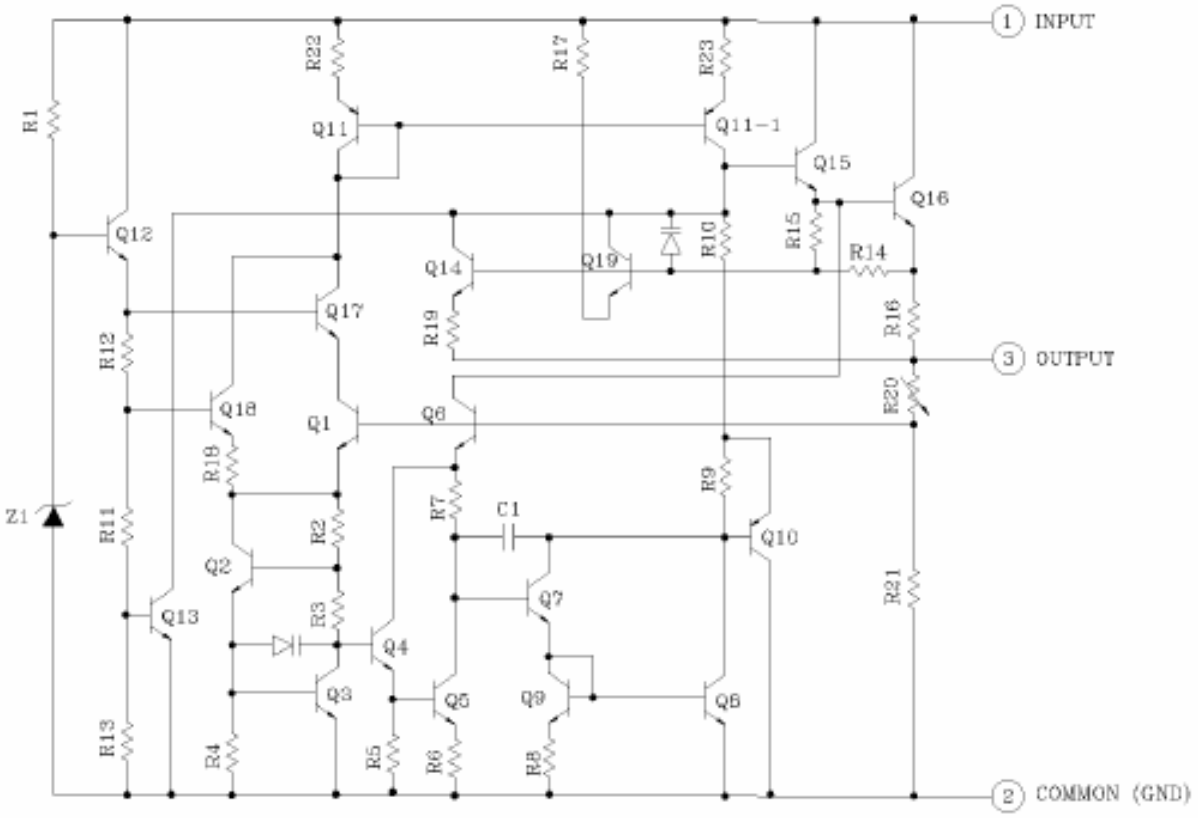


TO-220AB



TO-220IS

EQUIVALENT CIRCUIT



Typical Perfomance Characteristics

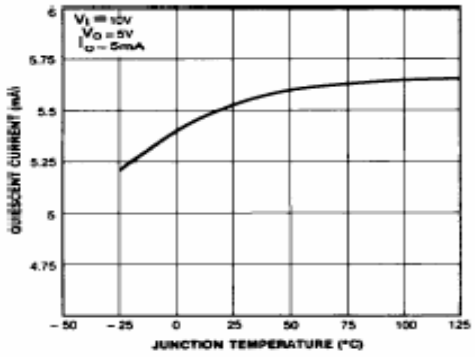


Figure 1. Quiescent Current

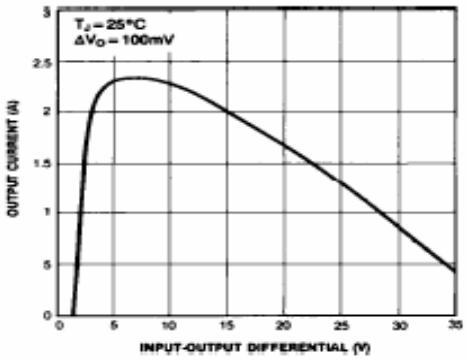


Figure 2. Peak Output Current

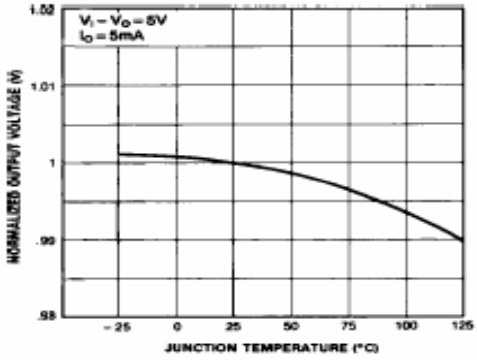


Figure 3. Output Voltage

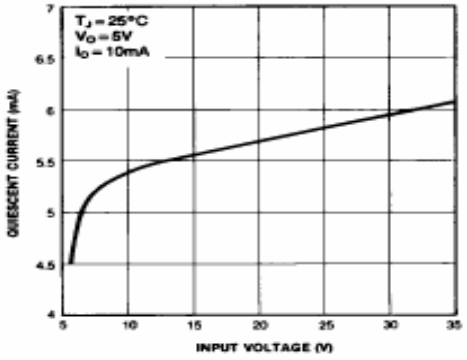


Figure 4. Quiescent Current

Vai a Robot “Pompiere”
Vai a Robot “Guardiano”
Vai a Possibili sviluppi
Torna all’indice

SOFTWARE ROBOT “POMPIERE” - BX-24

```
Option Explicit

dim dir1 as Byte
dim dir2 as Byte
dim dir3 as Byte
dim dir4 as Byte

dim ostacolo_sx as byte
dim ostacolo_dx as byte
dim fermo as byte
dim alzato as byte
dim vel_ff as integer
dim vel_bw as integer

Const ANT_Echo As Byte = 8                    ' Sonar Anteriore
Const ANT_Trig As Byte = 7
Dim Incremento As Integer
Dim Range As Integer
Dim Range_old As Integer

dim h as byte
dim m as byte
dim s as single


Public Sub Main()

Const timeout As Byte = 2

Const Pin_sx As Byte = 12
Const Pin_dx As Byte = 11
Const Pin_stop As Byte = 10
Const Pin_alzato As Byte = 9

Const GreenLED As Byte = 26
```

```
Const RedLED As Byte = 25
Const LEDon As Byte = 0
Const LEDoff As Byte = 1

' Define pins as input.
Call PutPin(17, bxInputPullup)
Call PutPin(18, bxInputPullup)
Call PutPin(19, bxInputPullup)
Call PutPin(20, bxInputPullup)

Call PutPin(ANT_Echo, bxInputTristate)
Call PutPin(ANT_Trig, bxOutputLow)

Call puttime(0,0,0.0)

'call tornacasa

do

 call gettime(h,m,s)
 if (m>timeout) and (alzato=0) then              'tornacasa dopo timeout minuti
   call tornacasa
 end if

 ostacolo_sx = getpin(Pin_sx)
 ostacolo_dx = getpin(Pin_dx)
 fermo = getpin(Pin_stop)
 alzato = getpin(Pin_alzato)

 Call PulseOut(ANT_Trig, 10, 1)
 Range_old=range
 Range = PulseIn(ANT_Echo, 1) \ 54              ‘lettura distanza sonar

 if (range<150) and (incremento>50) then
   incremento=20
 end if
```

```
 if (range>cint(csng(range_old)*0.9)) and  (range<cint(csng(range_old)*1.1)) then
  incremento=incremento+1
 else
  incremento=0
 end if

if alzato=0 then
  vel_ff=1350
  vel_bw=1550
  Call PutPin(RedLED, LEDoff)
else
   vel_ff=1350
   vel_bw=1550
   Call PutPin(GreenLED, LEDoff)
   Call PutPin(RedLED, LEDon)
end if


if fermo=1 then
   Call PutPin(GreenLED, LEDoff)
   Call PutPin(RedLED, LEDon)
else
   Call PutPin(RedLED, LEDoff)

   if (ostacolo_sx=0) and (ostacolo_dx=0) then
       Call PutPin(GreenLED, LEDon)
       Call avanti
   end if

   if (ostacolo_sx=1) and (ostacolo_dx=0) then
       Call PutPin(GreenLED, LEDoff)
         Call FreqOut(14, 3000, 3000, 50)
       Call gira_dx
   end if

   if (ostacolo_dx=1) and (ostacolo_sx=0) then
       Call PutPin(GreenLED, LEDoff)
```

```
          Call FreqOut(14, 2000, 2000, 50)
        Call gira_sx
    end if

    if (ostacolo_sx=1) and (ostacolo_dx=1) then
        Call PutPin(GreenLED, LEDoff)
          Call FreqOut(14, 5000, 5000, 50)
        Call turna
    end if

end if

loop

End Sub

public sub avanti()
'  debug.print "     GO !"

  call Pulseout (15, vel_ff-incremento, 1)
  call Pulseout (16, vel_ff-incremento, 1)

  call Pulseout (15, 1300, 1)
  call Pulseout (16, 1300, 1)

  Call Delay(0.07)

end sub

public sub gira_sx()
dim x as integer

  debug.print "     SX"

  for x=0 to 5
    call Pulseout (15, vel_ff-50, 1)
    call Pulseout (16, vel_bw, 1)
```

```
    call Delay(0.07)
  next
end sub

public sub gira_dx()
dim x as integer

  debug.print "     DX"

  for x=0 to 5
    call Pulseout (15, vel_bw, 1)
    call Pulseout (16, vel_ff-50, 1)
    call Delay(0.07)
  next

end sub


public sub turna()
dim x as integer

  debug.print "     TURNA"

  for x=0 to 5
  call Pulseout (15, vel_bw, 1)
  call Pulseout (16, vel_bw, 1)
  Call Delay(0.07)
  next

  for x=0 to 15
  call Pulseout (15, vel_ff-50, 1)
  call Pulseout (16, vel_bw, 1)
  Call Delay(0.07)
  next
end sub

public sub tornacasa()
```

```
dim k as integer
Const Pin_sx As Byte = 12
Const Pin_dx As Byte = 11

Const C As Integer = 785
Const D As Integer = 881
Const E As Integer = 989
Const G As Integer = 1176
Const r As Integer = 0

vel_ff=1380
vel_bw=1510

 debug.print "     TORNA A CASA"

 for k=0 to 50
 call pulseout (6, 1400, 1)      'alza bussola
 debug.print cstr(k)
 delay (0.02)
 next

 Call FreqOut(14, C, C, 2.0E-1)
 Call FreqOut(14, E, E, 2.0E-1)
 Call FreqOut(14, G, G, 2.0E-1)
 Call FreqOut(14, C, C, 4.0E-1)
 Call FreqOut(14, r, r, 1.0E-1)
 Call FreqOut(14, G, G, 2.0E-1)
 Call FreqOut(14, C, C, 1.0)

  vel_ff=1300
  vel_bw=1600

 do
   'ostacolo_sx = getpin(Pin_sx)
   'ostacolo_dx = getpin(Pin_dx)

   call pulseout (6, 1400, 1)                    'tiene alzata bussola
```

```
'   Call PulseOut(ANT_Trig, 10, 1)
'   Range_old=range
'   Range = PulseIn(ANT_Echo, 1) \ 54            ' lettura distanza sonar

'   if (range<150) and (incremento>50) then
'     incremento=20
'   end if

'   if (range>cint(csng(range_old)*0.9)) and  (range<cint(csng(range_old)*1.1)) then
'     incremento=incremento+1
'   else
'     incremento=0
'   end if
'           debug.print "ostacolo_SX "; cstr(ostacolo_sx);
 '          debug.print "ostacolo_DX "; cstr(ostacolo_dx);

'   if (ostacolo_sx=0) and (ostacolo_dx=0) then
      dir1 = GetPin(17)
      dir2 = GetPin(18)                              q 'legge bussola
      dir3 = GetPin(19)
      dir4 = GetPin(20)

       if dir1=0 then
         for k=0 to 5
          call Pulseout (15, 1200, 1)                       'gira a DX
          call Pulseout (16, 1600, 1)
          call Delay(0.07)
          debug.print "gira a DX"
                next
               end if

      if dir2=0 then
        for k=0 to 5
         call Pulseout (15, 1300, 1)                        'robot dritto
         call Pulseout (16, 1300, 1)
         Call Delay(0.07)
```

```
          debug.print "dritto"
                    next
        end if

        if dir3=0 then
          for k=0 to 5
            call Pulseout (15, 1600, 1)                    'gira a SX
            call Pulseout (16, 1200, 1)
            call Delay(0.07)
            debug.print "gira a SX"
          next
                  end if

        if dir4=0 then
          for k=0 to 25                                    'inverte rotta a 180°
          call Pulseout (15, 1200, 1)
          call Pulseout (16, 1600, 1)
          Call Delay(0.07)
          next
          debug.print "inverte rotta"
                  end if
'   end if

'   if (ostacolo_sx=1) and (ostacolo_dx=0) then
'         Call FreqOut(14, 3000, 3000, 50)
'        Call gira_dx
'   end if

'   if (ostacolo_dx=1) and (ostacolo_sx=0) then
'         Call FreqOut(14, 2000, 2000, 50)
'        Call gira_sx
'   end if

'   if (ostacolo_sx=1) and (ostacolo_dx=1) then
'         Call FreqOut(14, 5000, 5000, 50)
'
'           for k=0 to 5
```

```
'           call Pulseout (15, vel_bw, 1)          'arretra
'           call Pulseout (16, vel_bw, 1)
'           Call Delay(0.07)
'           next

'       if dir4=0 then
'           for k=0 to 15
'           call Pulseout (15, vel_ff-50, 1)       'gira a DX
'           call Pulseout (16, vel_bw, 1)
'           Call Delay(0.07)
'           next
'       end if

'       if dir1=0 then
'           for k=0 to 15
'           call Pulseout (15, vel_bw, 1)          'gira a SX
'           call Pulseout (16, vel_ff-50, 1)
'           Call Delay(0.07)
'           next
'       end if

'   end if
loop

 arrivato:

 for k=1 to 10
  call Pulseout (6, 0.7E-3, 1)          'abbassa la bussola
  debug.print cstr(k)
  delay (0.1)
 next

end sub
```

Vai a Robot “Pompiere”
Vai a Robot “Guardiano”
Vai a Possibili sviluppi
Torna all’indice

SOFTWARE ROBOT “POMPIERE” - OOPic

```
Dim sens_r As New oIRPD
Dim sens_l As New oIRPD
Dim ostacolo as New obyte
Dim Pyro As New oi2c
Dim x as new obyte
Dim y As New oByte
Dim trovato as new obit
Dim alzato as new obyte
Dim pin_alzato as new oDio1
Dim pin_sx as new oDio1
Dim pin_dx as new oDio1
Dim pin_stop as new oDio1
Dim pos as new obyte
Dim verso_pos as new obit
Dim tmax as new obyte
Dim tmin as new obyte
Dim tref as new obyte
Dim tempa as new obyte
Dim tempb as new obyte
Dim tempc as new obyte
Dim tempd as new obyte
Dim tempe as new obyte
Dim tempf as new obyte
Dim tempg as new obyte
Dim temph as new obyte
Dim tempi as new obyte
Dim A As New oFreq
Dim Led_rosso As New oDio1
Dim Led_giallo As New oDio1
Dim Led_verde As New oDio1
Dim S as new oServo
Dim T as new oServo
Dim O as new oServo
Dim G as new oDio1
Dim centro As New oByte
```

```
Dim corsa As New oByte
Dim limite_sx As New oByte
Dim limite_dx As New oByte

Sub main()

Call SetUp

Call esplora
if trovato then goto inizio

pin_dx=0:pin_sx=0:pin_stop=1
OOPic.delay = 200
pin_stop=0

A.Beep(60000,10)
A.Beep(61000,10)
A.Beep(63000,10)

do

  call legge_temp

  call trova_calore              'setta trovato=1
  Led_giallo=trovato

  if trovato=0 then call naviga

  if trovato=1 then

    inizio:
    if (tmax>80) then                                  'ferma robot
      pin_stop=1
    else
      pin_stop=0
    end if
```

```
 if (tmax-tmin>30) and (alzato=0) then                    'alza ventola
       pin_dx=0:pin_sx=0
  if tmax>50 then pin_stop=1 else pin_stop=0
  alzato=1: Led_rosso=alzato:  pin_alzato=alzato
  call alza
  call legge_temp
 end if

 if (tmax>80) and (alzato=1) then                         'spegne fiamma
  pin_stop=1
  call spegne
  pin_stop=0
  pin_dx=1:pin_sx=1     'arretra e si allontana
 end if

 if pos>20 then pos=pos - 2:Led_verde=1:pin_dx=1:pin_sx=0    'orienta a sx
 if pos<10 then pos=pos + 2:Led_verde=1:pin_dx=0:pin_sx=1    'orienta a dx
 if pos<5 then pos=5: verso_pos=1
 if pos>26 then pos=26: verso_pos=0
 Pyro.location=0:  Pyro = pos

end if


if (tmax-tmin<20) and (alzato=1) then                  'controlla se persa fiamma
  y=0
  pin_stop=1
 do
  if verso_pos=0 then pos=pos - 3                     'spazzola l'orizzonte
  if verso_pos=1 then pos=pos + 3
  if pos<4 then pos=4: verso_pos=1: y=y + 1
  if pos>28 then pos=28: verso_pos=0: y=y + 1
  Pyro.location=0:  Pyro = pos
  call legge_temp
  call trova_calore                          'setta trovato=1
  if trovato=1 then goto inizio
 loop while y<3
```

```
  pin_stop=1
  call abbassa                                    'abbassa ventola
  alzato=0: Led_rosso=alzato:  pin_alzato=alzato
  if centro>0 then pin_dx=1:pin_sx=1              'arretra e si allontana
  pin_stop=0
 end if

loop

End Sub



Sub SetUp()

 Pyro.Node = 104
 Pyro.Width = cv8bit       'Control Info is 1-byte
 Pyro.Mode = cv10bit       'I2C mode is 10-Bit Addressing
 Pyro.NoInc = cvTrue       'Don't increment the Location.

 tmax=0
 pos=15
 verso_pos=0
 Pyro.location=0
 Pyro = pos
 centro=0
 y=0

 S.IOLine = 31
 S.Center = 33
 O.IOLine = 30
 O.Center = 30
 T.IOLine = 29
 T.Center = 30
 S.Operate = cvFalse
 O.Operate = cvFalse
```

```
T.Operate = cvFalse

G.IOline=8                    'comando ventola
G.Direction=cvOutput
G=0


Led_verde.IOLine = 5                  'led board
Led_giallo.IOLine = 6
Led_rosso.IOLine = 7
Led_verde.Direction = cvOutput
Led_giallo.Direction = cvOutput
Led_rosso.Direction = cvOutput

sens_r.IOLineL = 10                'sensore IRPD dx
sens_r.IOLineR = 11
sens_r.IOLineS = 12
sens_r.InvertIn = cvTrue
sens_r.Operate = 1

sens_l.IOLineL = 13                'sensore IRPD sx
sens_l.IOLineR = 14
sens_l.IOLineS = 15
sens_l.InvertIn = cvTrue
sens_l.Operate = 1

pin_sx.IOline=1                   'comunica ostacolo a sx
pin_sx.Direction=cvOutput
pin_sx=0

pin_dx.IOline=2                   'comunica ostacolo a dx
pin_dx.Direction=cvOutput
pin_dx=0

pin_stop.IOline=3                 'comunica stop motori
pin_stop.Direction=cvOutput
pin_stop=0
```

```
  pin_alzato.IOline=4                'comunica ventola alzata
  pin_alzato.Direction=cvOutput
  alzato=0
  pin_alzato=alzato

End Sub

Sub alza()
  A.Beep(60000,10)
  A.Beep(61000,10)
  A.Beep(63000,10)

 S.Operate = cvTrue
 O.Operate = cvTrue
 T.Operate = cvTrue

 for x= 0 to 54
 S=55-x/2 : O=27: T=57-x/2:  OOPic.delay = 0.5
 S=55-x/2 : O=27: T=57-x/2:  OOPic.delay = 0.5
 next x

 for x= 0 to 50
 S=28 : O=27 : T=30 :  OOPic.delay = 0.5
 next x

  A.Beep(60000,10)
  A.Beep(61000,10)
  A.Beep(63000,10)

end sub

Sub abbassa()
  A.Beep(60000,10)
  A.Beep(61000,10)
  A.Beep(63000,10)
```

```
S.Operate = cvTrue
O.Operate = cvTrue
T.Operate = cvTrue

for x= 0 to 30
S=28 : O=30 : OOPic.delay = 0.5
next x

for x= 0 to 51
S=28+x/3 : O=30: T=37+x/2 :  OOPic.delay = 0.5
S=28+x/3 : O=30: T=37+x/2 :  OOPic.delay = 0.5
S=28+x/3 : O=30: T=37+x/2 :  OOPic.delay = 0.5
next x

for x= 0 to 30
S=45+x/3 : O=30-x/10 : T=57:  OOPic.delay = 0.5
S=45+x/3 : O=30-x/10 : T=57:  OOPic.delay = 0.5
S=45+x/3 : O=30-x/10 : T=57:  OOPic.delay = 0.5
next x

 A.Beep(60000,10)
 A.Beep(61000,10)
 A.Beep(63000,10)

 S.Operate = cvFalse
 O.Operate = cvFalse
 T.Operate = cvFalse

End Sub

Sub spegne()

  A.Beep(47000,5)
  A.Beep(57000,5)

  S.Operate = cvTrue
  O.Operate = cvTrue
```

```
T.Operate = cvTrue

do
  x=0
  call legge_temp
  if (tempb>tref) or (tempc>tref) then pos=pos + 1: x=1
  if (temph>tref) or (tempi>tref) then pos=pos - 1: x=1
  if pos<2 then pos=2: verso_pos=1
  if pos>28 then pos=28: verso_pos=0
  Pyro.location=0:  Pyro = pos
loop while x=1

if (pos>=0)  and (pos<10) then centro=12: corsa=20
if (pos>=10) and (pos<15) then centro=26: corsa=12
if (pos>=15) and (pos<20) then centro=32: corsa=10
if (pos>=20) and (pos<25) then centro=34: corsa=12
if (pos>=25) and (pos<31) then centro=48: corsa=20

limite_sx=centro-corsa
limite_dx=centro+corsa

for y=0 to 2

   x=1
   do until O.value<limite_sx
         G=1
     S=28 : T=36 + tmax/25 + y
     O.value=O.value-x
     OOPic.delay = 3
   loop

   do until O>limite_dx
         G=1
     S=28 : T=36 + y
     O.value=O.value+x
     OOPic.delay = 3
   loop
```

```
  next y

  G=0
  G=0
  G=0


  A.Beep(47000,5)
  A.Beep(57000,5)

End Sub


Sub legge_temp()

  Pyro.location=1: tempa=pyro      'temp ambiente
  Pyro.location=2: tempb=pyro
  Pyro.location=3: tempc=pyro
  Pyro.location=4: tempd=pyro
  Pyro.location=5: tempe=pyro      'lettura array sensore
  Pyro.location=6: tempf=pyro
  Pyro.location=7: tempg=pyro
  Pyro.location=8: temph=pyro
  Pyro.location=9: tempi=pyro

  tmax=0

  if tempb>tmax then tmax=tempb
  if tempc>tmax then tmax=tempc       'calcolo temp max
  if tempd>tmax then tmax=tempd
  if tempe>tmax then tmax=tempe
  if tempf>tmax then tmax=tempf
  if tempg>tmax then tmax=tempg
  if temph>tmax then tmax=temph
  if tempi>tmax then tmax=tempi
```

```
  tmin=1000
  if tempa<tmin then tmin=tempa
  if tempb<tmin then tmin=tempb
  if tempc<tmin then tmin=tempc
  if tempd<tmin then tmin=tempd       'calcolo temp min misurata
  if tempe<tmin then tmin=tempe
  if tempf<tmin then tmin=tempf
  if tempg<tmin then tmin=tempg
  if temph<tmin then tmin=temph
  if tempi<tmin then tmin=tempi

End Sub

Sub ctrl_ostacoli()

 ostacolo=0

  if (sens_r=2) then ostacolo=2                'ostacolo a dx
  if (sens_l=1) then ostacolo=1                'ostacolo a sx
  if (sens_r=3) then ostacolo=3                'ostacolo a dx
  if (sens_l=3) then ostacolo=3                'ostacolo al centro
  if (sens_r=1) and (sens_l=2) then ostacolo=3    'ostacolo al centro
  if (sens_l=3) and (sens_r=3) then ostacolo=3    'ostacolo al centro
  if (sens_l=0) and (sens_r=0) then ostacolo=0    'nessun ostacolo

End Sub


Sub trova_calore()

 tref=tmin+10          ' cerca dove ci sono +10

 if (tempb>tref) or (tempc>tref) or (tempd>tref) or (tempe>tref) or (tempf>tref) or
(tempg>tref) or (temph>tref) then
    trovato=1
 else
    trovato=0
```

```
  end if

End Sub


Sub naviga()
  if alzato=0 then                              'scansione spazio
'   if y>100 then call esplora
   if trovato=1 then goto valle
   if verso_pos=0 then pos=pos - 4
   if verso_pos=1 then pos=pos + 4
   if pos<7 then pos=7: verso_pos=1
   if pos>26 then pos=26: verso_pos=0
   Pyro.location=0:  Pyro = pos

   call ctrl_ostacoli                           'setta ostacolo=0,1,2,3
   if ostacolo=0 then Led_verde=1: pin_dx=0:pin_sx=0     'viaggia dritto
   if ostacolo=1 then Led_verde=0: pin_dx=0:pin_sx=1     'ostacolo a sx
   if ostacolo=2 then Led_verde=0: pin_dx=1:pin_sx=0     'ostacolo a dx
   if ostacolo=3 then Led_verde=0: pin_dx=1:pin_sx=1     'ostacolo al centro
   y=y + 1
  end if

  if alzato=1 then
    Led_verde=1: pin_dx=0:pin_sx=0                      'viaggia dritto
    if pos>20 then pos=pos - 1:Led_verde=1:pin_dx=1:pin_sx=0    'orienta a sx
    if pos<10 then pos=pos + 1:Led_verde=1:pin_dx=0:pin_sx=1    'orienta a dx
    if verso_pos=0 then pos=pos - 2
    if verso_pos=1 then pos=pos + 2
    if pos<7 then pos=7: verso_pos=1
    if pos>26 then pos=26: verso_pos=0
    Pyro.location=0:  Pyro = pos
  end if

valle:

End Sub
```

```
Sub esplora()

for x=0 to 55
  pin_dx=1:pin_sx=0
  pin_stop=0

  if verso_pos=0 then pos=pos - 1                    'esplora a 180°
  if verso_pos=1 then pos=pos + 1
  if pos<13 then pos=13: verso_pos=1
  if pos>18 then pos=18: verso_pos=0
  Pyro.location=0:  Pyro = pos

  call legge_temp
  call trova_calore
  Led_giallo=trovato
  if trovato=1 then goto dopo
next x

dopo:
y=0

End Sub
```

Vai a Robot “Pompiere”
Vai a Robot “Guardiano”
Vai a Possibili sviluppi
Torna all’indice

SOFTWARE ROBOT “GUARDIANO” - BX-24

```
Option Explicit

dim sensoreSX as single
dim sensoreDX as single


const acceso as byte = 1
const spento as byte = 0

'****************************
' ASCII values
'****************************
Const ASCII_CR    As Byte = 13



'****************************
' serial communication buffers
'****************************
Dim Com3InBuffer(1 To 48)   As Byte ' input buffer for all COM3 devices
Dim Com3OutBuffer(1 To 48)  As Byte ' output buffer for all COM3 devices
Dim servo0 as String * 20                   'base
Dim servo3 as String * 20                   'snodo
Dim servo6 as String * 20                   'camera
Dim servo4 as String * 20                   'perno
'****************************


Public Sub Main()

dim k as integer

' open serial communication queues
   Call OpenQueue(Com3InBuffer, 48)
   Call OpenQueue(Com3OutBuffer, 48)
```

```
k=0
do

k=k+1

Call Ostacoli

if (sensoreDX<>0.0) and (sensoreSX <>0.0) then
  debug.print " OSTACOLO DAVANTI!"
  Call indietro
  call delay(2.0)
  Call girasx
end if

if (sensoreDX = 0.0) and (sensoreSX = 0.0) then
  debug.print " LIBERO!"
  Call avanti
end if


if (sensoreDX<>0.0) and (sensoreSX =0.0)  then
  debug.print " OSTACOLO A DX!"
  Call girasx
end if


if (sensoreSX<>0.0) and (sensoreDX = 0.0)   then
  debug.print " OSTACOLO A SX!"
  Call giradx
end if


if k=50 then
k=0
Call curioso
Call delay(3.0)
Call riposo
```

```
        Call Delay(6.0)
        Call intermedio
        Call Delay(8.0)
    Call eretto
        Call Delay(8.0)
        Call erettodestra
        Call Delay(8.0)
        Call erettosinistra
        Call Delay(8.0)
        Call Riposo
    Call delay(5.0)
end if


loop
End sub

Private Sub Ostacoli()

call Putpin(13, acceso)
sensoreSX=  csng(pulsein(15, 1))
call Putpin(13, spento)


call Putpin(14, acceso)
sensoreDX=  csng(pulsein(15, 1))
call Putpin(14, spento)

End sub



Public Sub avanti()                    'inizio subroutine "avanti"(muove il robot in
avanti)
dim k as integer

for k=1 to 5
```

```
call pulseout (11, 1350, 1)
call pulseout (12, 1350, 1)
call delay(0.02)
next
end sub                          'fine subroutine "avanti"

Public sub indietro()            'inizio subroutine "indietro"(muove il robot
dim k as integer                 'all'indietro)

for k=1 to 10
call pulseout (11, 1550, 1)
call pulseout (12, 1550, 1)
call delay(0.02)
next
end sub                          'fine subroutine "indietro"

Public sub girasx()              'inizio subroutine "girasx"(fa girare il robot a
sinistra)
dim k as integer

for k=1 to 5
call pulseout (11, 1350, 1)
call pulseout (12, 1550, 1)
call delay(0.02)
next
call delay(2.0)
end sub                          'fine subroutine "girasx"

Public sub giradx()              'inizio subroutine "giradx"(fa girare il robot a
destra)
dim k as integer

for k=1 to 5
call pulseout (11, 1550, 1)
call pulseout (12, 1350, 1)
call delay(0.02)
next
```

```
call delay(2.0)
end sub                                          'fine subroutine "giradx"
'*********************************************************************
' Open Com3
'*********************************************************************
Private Sub curioso()
   Call DefineCom3(0, 8, bx0000_1000) 'input port 0, output port 6 no inverted, no parity, 8
data
   Call OpenCom(3, 9600, Com3InBuffer, Com3OutBuffer)
   Debug.print
   Debug.print "----------"
   Debug.print "Aperta COM3"
        Call freqout(14,2000,2000,50)
End Sub
'*********************************************************************
' Send a command to Curiosone
'*********************************************************************
Private Sub Riposo()
        servo0="#0 p2390 t5000"
        servo3="#3 p590 t5000"
        servo6="#6 p1600 t5000"
        servo4="#4 p1500 t5000"
Debug.print "riposo"
Call freqout(14,2000,2000,50)
Call PutQueueStr(Com3OutBuffer,servo0 & Chr (ASCII_CR))
Call PutQueueStr(Com3OutBuffer,servo3 & Chr (ASCII_CR))
Call PutQueueStr(Com3OutBuffer,servo6 & Chr (ASCII_CR))
Call PutQueueStr(Com3OutBuffer,servo4 & Chr (ASCII_CR))
End Sub

Private Sub intermedio()
        servo0="#0 p1500 t5000"
        servo3="#3 p1750 t5000"
        servo6="#6 p1500 t5000"
        servo4="#4 p1500 t5000"
Call PutQueueStr(Com3OutBuffer,servo0 & Chr (ASCII_CR))
Call PutQueueStr(Com3OutBuffer,servo3 & Chr (ASCII_CR))
```

```
Call PutQueueStr(Com3OutBuffer,servo6 & Chr (ASCII_CR))
Call PutQueueStr(Com3OutBuffer,servo4 & Chr (ASCII_CR))
Debug.print "intermedio"
Call freqout(14,2000,2000,50)
End Sub

Private Sub intermediodestra()
        servo0="#0 p1500 t5000"
        servo3="#3 p1750 t5000"
        servo6="#6 p1500 t5000"
        servo4="#4 p2300 t5000"
Call PutQueueStr(Com3OutBuffer,servo0 & Chr (ASCII_CR))
Call PutQueueStr(Com3OutBuffer,servo3 & Chr (ASCII_CR))
Call PutQueueStr(Com3OutBuffer,servo6 & Chr (ASCII_CR))
Call PutQueueStr(Com3OutBuffer,servo4 & Chr (ASCII_CR))
Debug.print "intermedio"
Call freqout(14,2000,2000,50)
End Sub

Private Sub intermediosinistra()
        servo0="#0 p1500 t5000"
        servo3="#3 p1750 t5000"
        servo6="#6 p1500 t5000"
        servo4="#4 p750 t5000"
Call PutQueueStr(Com3OutBuffer,servo0 & Chr (ASCII_CR))
Call PutQueueStr(Com3OutBuffer,servo3 & Chr (ASCII_CR))
Call PutQueueStr(Com3OutBuffer,servo6 & Chr (ASCII_CR))
Call PutQueueStr(Com3OutBuffer,servo4 & Chr (ASCII_CR))
Debug.print "intermedio"
Call freqout(14,2000,2000,50)
End Sub

Private Sub eretto()
        servo0="#0 p1460 t5000"
        servo3="#3 p2380 t5000"
        servo6="#6 p800 t5000"
        servo4="#4 p1500 t5000"
```

```
Call PutQueueStr(Com3OutBuffer,servo0 & Chr (ASCII_CR))
Call PutQueueStr(Com3OutBuffer,servo3 & Chr (ASCII_CR))
Call PutQueueStr(Com3OutBuffer,servo6 & Chr (ASCII_CR))
Call PutQueueStr(Com3OutBuffer,servo4 & Chr (ASCII_CR))
Debug.print "eretto"
Call freqout(14,2000,2000,50)
End Sub

Private Sub erettodestra()
        servo0="#0 p1460 t5000"
        servo3="#3 p2380 t5000"
        servo6="#6 p800 t5000"
        servo4="#4 p2300 t5000"
Call PutQueueStr(Com3OutBuffer,servo0 & Chr (ASCII_CR))
Call PutQueueStr(Com3OutBuffer,servo3 & Chr (ASCII_CR))
Call PutQueueStr(Com3OutBuffer,servo6 & Chr (ASCII_CR))
Call PutQueueStr(Com3OutBuffer,servo4 & Chr (ASCII_CR))
Debug.print "eretto"
Call freqout(14,2000,2000,50)
End Sub

Private Sub erettosinistra()
        servo0="#0 p1460 t5000"
        servo3="#3 p2380 t5000"
        servo6="#6 p800 t5000"
        servo4="#4 p750 t5000"
Call PutQueueStr(Com3OutBuffer,servo0 & Chr (ASCII_CR))
Call PutQueueStr(Com3OutBuffer,servo3 & Chr (ASCII_CR))
Call PutQueueStr(Com3OutBuffer,servo6 & Chr (ASCII_CR))
Call PutQueueStr(Com3OutBuffer,servo4 & Chr (ASCII_CR))
Debug.print "eretto"
Call freqout(14,2000,2000,50)
End Sub
```

Vai a Robot “Pompiere”
Vai a Robot “Guardiano”
Vai a Possibili sviluppi
Torna all’indice

TELECAMERA WIRELESS

Questa Telecamera a colori e' dotata di trasmittente a 2.4Ghz che rende possibile il libero posizionamento in qualsiasi parte dell'ambiente, ottimo per circuiti di telesorveglianza, antifurti, telepresenza ma anche come teleguida di robot come nel nostro caso.
Le dimensioni molto compatte (23x23x23mm) la rendono facilmente installabile, anche su robot.

TRASMETTITORE

Il trasmettitore e' incluso nel piccolo contenitore della camera, anche un piccolo microfono trova posto nello stesso . Il modulo e' provvisto di presa DC e un'antenna;esso consente di avere immagini chiare e nitide fino a distanze di circa 100 metri all'aperto. La trasmittente ha una potenza di 10mW ed il Kit e' omologato con la normativa CE.

RICEVITORE

Il ricevitore e è provvisto di uscite video e audio, dispone di 4 canali e antenna. Una serie di LED posti sul pannello frontale permettono di sapere canale in uso.

IL KIT INCLUDE:

Telecamera con trasmettitore 2.4Ghz
Ricevitore 2.4Ghz
Alimentatore Telecamera
Alimentatore Ricevitore
Batteria 9V per camera
Cavo Audio-Video per collegare il Ricevitore alla TV o PC
Piccolo supporto per la camera

SPECIFICHE TECNICHE TELECAMERA:

Sistema: PAL
Pixel Effettivi: 628x582
Risoluzione: 380 linee
Sincronizzazione: interna
Fotogrammi/s: 25
Illuminazione minima: 1.5 Lux
Bilanciamento del bianco: ATW
Consumo: 80mA max
Temperatura di esercizio: -20+50°C

SPECIFICHE TECNICHE TRASMETTITORE:

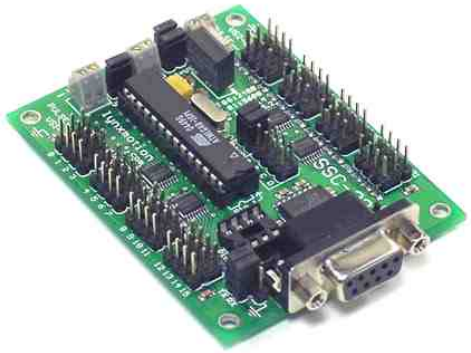
Frequenza: Ch1=2.414 Mhz , Ch2=2.432 Mhz, Ch3=2.450 Mhz, Ch4=2.468 Mhz
Potenza: 10dBm
Tipo di modulazione: FM

SPECIFICHE TECNICHE RICEVITORE:

Frequenza: Ch1=2.414 Mhz , Ch2=2.432 Mhz, Ch3=2.450 Mhz, Ch4=2.468 Mhz
Sensibilita': -85dBm
Tipo di demodulazione: FM
Stabilita' frequenza: +/- 100Khz
Consumo: 250mA
Peso: 200gr
Dimensioni:78 x 92 x 23 mm
Temperatura di esercizio: -20+50°C

Vai a Robot “Pompiere”
Vai a Robot “Guardiano”
Vai a Possibili sviluppi
Torna all’indice

CONTROLLORE SERVI (SSC-32)

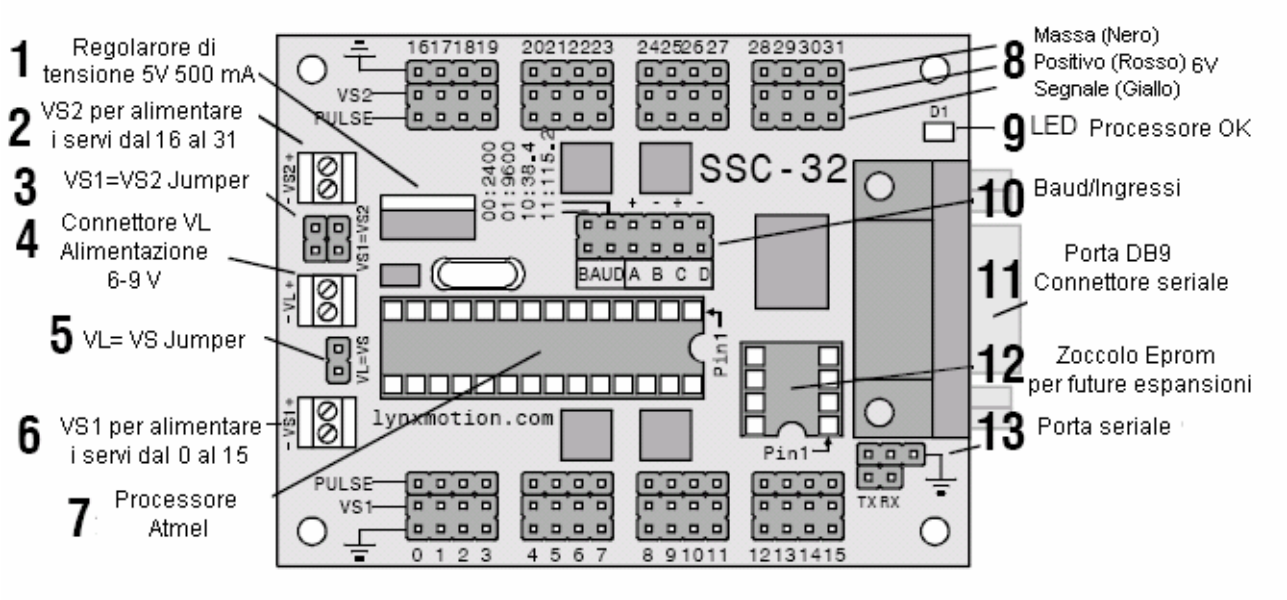


L' SSC-32 (controllore servi seriale) è un piccolo controllore servi preassemblato. E' dotato di un' alta risoluzione (1us) per ottenere un posizionamento accurato e movimenti graduali. Il range di durata dell'impulso di controllo per un movimento di 180° del servo va da 0,5 a 2,50ms. Il controllo del movimento può essere a risposta immediata, a velocità controllata, a movimento temporizzato o una combinazione di questi. Il controllore è in grado di gestire il movimento di più servi contemporaneamente anche se essi devono eseguire movimenti differenti. La posizione ed il movimento del servo può essere visualizzata a computer. Sono presenti quattro ingressi digitali che possono essere usati anche come ingressi analogici, tre ingressi di alimentazione, ed una posta seriale RS-232 per comunicare con un PC.

SPECIFICHE

Microcontrollore = Atmel ATMEGA8-16PI
EEPROM = 24LC32P (Non supportata in questa versione)
Frequenza = 14.75 MHz
Ingresso seriale DB9F= RS-232 o TTL, 2400, 9600, 38.4k, 115.2k, N81
Uscita = 32 (Servo o TTL)
Ingressi = 4 (Static o Latching)
Consumo di corrente = 31mA (solo scheda)
Controllo servi = fino a 32 connessi direttamente
Servi supportati = Futaba o Hitec
Rotazione servo = ~170°
Tipo di controllo dei servi= immediato, temporizzato, di velocità o una combinazione di questi.
PC board size = 7,6 cm x 5,8 cm

CARATTERISTICHE

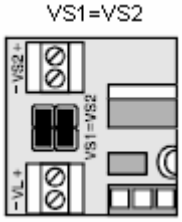


1) Il regolatore a bassa caduta di tensione permette di fornire i 5vdc necessari al circuito anche avendo solo 5,5vdc in ingresso (tensione di ingresso MAX 9vdc).Questo è importante quando si alimenta il robot con una batteria. Il regolatore è in grado di erogare fino a 500mA ma è consigliabile non superare i 250mA per evitare un surriscaldamento.

2) Questo connettore permette di alimentare i canali servo da 16 a 31: per servi normali è necessario applicare una tensione compresa tra 4.8vdc e 7.2vdc, mentre per i micro servi essa può variare tra 4.8vdc e 6vdc.

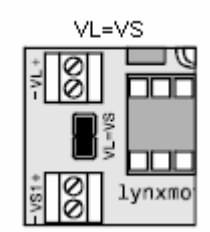
Board	Input
VS2+	RED
VS2 -	BLACK

3) Inserire i due jumper per connettere VS1 A VS2: questa opzione si utilizza quando tutti i servi vengono alimentati tramite la stessa batteria.



4) Questo è l'ingresso per l'alimentazione dell'integrato e di qualunque componente sia collegato alla linea dei 5vcd sulla scheda ed è detta Logic Voltage (VL).

5) Questo jumper permette di alimentare il microcontrollore e la circuiteria di supporto con l'alimentazione dei servi: per far questo è richiesta un'alimentazione di almeno 6vdc. Se il microcontrollore dovesse resettarsi quando i servi sono in funzione è necessario togliere il jumper ed utilizzare la VL.



6) Questo connettore permette di alimentare i canali servo da 0 a 15: per servi normali è necessario applicare una tensione compresa tra 4.8vdc e 7.2vdc, mentre per i micro servi essa può variare tra 4.8vdc e 6vdc.

Board	Input
VL1+ VL1 -	RED BLACK

7) Questo è lo zoccolo dove vi è inserito il microcontrollore Atmel. Inserirlo in modo corretto rispettando la piedinatura.

8) Sono le porte dove si connettono i servi o altri dispositivi. -Attenzione- Prima di collegare un dispositivo a queste porte scollegare sempre l'alimentazione!

9) Led che indica il corretto funzionante del processore. Quando viene accesa l'alimentazione esso si accende e rimane acceso finchè riceve un segnale valido dalla porta seriale. Successivamente si spegne e lampeggia ogni volta che legge dei dati sulla seriale.

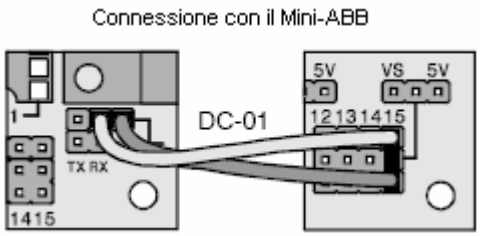
10) I due ingressi di Baud permettono di configurare il Baud rate.Gli ingressi A,B,C,D hanno delle resistenze di pull-up da 50 KΩ.

Jumpers	Baud Rate
0 0	2400
0 1	9600
1 0	38.4k
1 1	115.2k

11) Porta seriale per collegare la scheda al PC

12) Zoccolino per EEPROM 8 pin.

13) Porta seriale TTL per la comunicazione con il Mini-ABB.Inserire i due jumper come mostrato nella figura sotto per abilitare la porta seriale DB9.



Vai a Robot “Pompiere”
Vai a Robot “Guardiano”
Vai a Possibili sviluppi
Torna all’indice

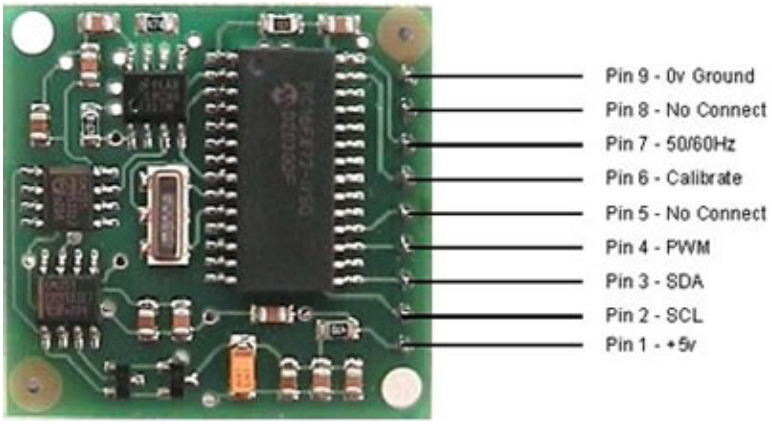
BUSSOLA ELETTRONICA

La bussola elettronica fornisce la direzione verso la quale si dirige il Robot. Per la comunicazione con il microcontrollore usa una uscita PWM (scelta fatta da noi) oppure una interfaccia I2C (non utilizzata) disponibile tramite i pin 2 e 3. E' molto piccola e precisa.

Caratteristiche Tecniche:	
Tensione Operativa	5V
Corrente Operativa Tipica	20mA
Risoluzione	0.1°
Accuratezza	3-4 ° circa, dopo la calibrazione
Uscita 1	Durata impulso da 1mS a 37mS con incrementi da 0.1mS
Uscita 2	Interfaccia I2C, 0-255 e 0-3599 Velocità SCL fino a 1MHz
Dimensioni	32 x 35 mm

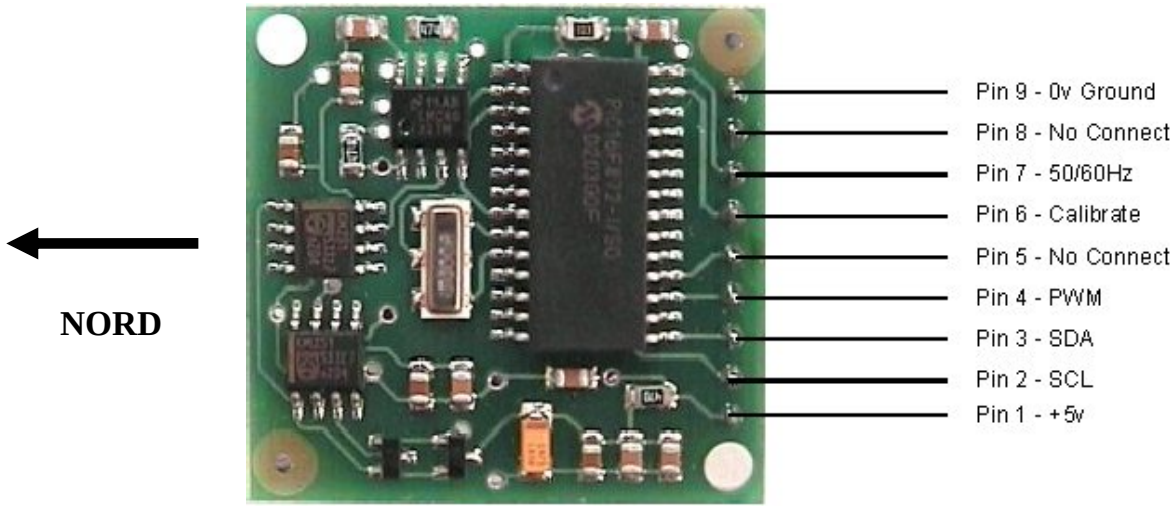
Questa bussola è stata progettata specificatamente per l'uso nella robotica e come supporto nella navigazione. Essa utilizza i sensori di campo magnetico Philips KMZ51 che sono abbastanza sensibili da rilevare il campo magnetico terrestre. L'uscita proveniente da due di questi sensori permette di calcolare la direzione delle componenti orizzontali del campo magnetico della Terra.

CONNESSIONI



La bussola richiede un'alimentazione di 5V ad una corrente nominale di 15mA. La larghezza dell'impulso PWM in uscita dalla bussola rappresenta l'angolo di scostamento in senso orario rispetto al nord: essa varia da 1mS (0°) a 36,99mS(359,9°), quindi una variazione di 100uS corrisponde ad uno scostamento di 1°. Il segnale va a livello basso per 65mS tra un impulso e l'altro così che la durata del ciclo è pari a 65mS più la durata dell'impulso. L'impulso viene generato nel processore (PIC) da un timer a 16 bit che dà una risoluzione di 1uS, comunque è sconsigliato misurare scostamenti inferiori a 0,1° (10uS). Se si utilizza l'uscita PWM assicurarsi di connettere i pin I2C, SCL e SDA, all'alimentazione (5V) dato che essi non dispongono di resistori di pull-up.
Per evitare le interferenze dovute all'alimentazione di rete (che possono provocare scostamenti nella misurazione di circa 1,5°) si utilizza il pin7 a cui si può applicare un segnale con frequenza 50 o 60Hz a seconda del disturbo che si vuole eliminare: questo permette di ridurre a 0,2° l'errore.
I pin 5 e 8 non vengono connessi..
Il pin 6 serve per calibrare la bussola ed ha un resistore di pull-up on board così che possa essere lasciato disconnesso dopo la calibrazione.

CALIBRAZIONE



Prima di utilizzare per la prima volta la bussola bisogna procedere alla calibrazione. I dati della calibrazione vengono memorizzati nella EEPROM all'interno del PIC16F872. Una volta calibrata non è necessario ripetere questa operazione ogni volta che si spegne e riaccende il circuito.

Di seguito sono riportati registri e corrispondenti funzioni utilizzati nella calibrazione tramite I2C.

Registri	**Funzioni**
0	Versione del sofware
1	Formato dati rilevati :8bit, da 0-255 per un giro completo
2,3	Formato dati rilevati : 16bit, da. 0-3599 per un giro completo, rappresentanti i gradi da 0 a 360.
4,5	Segnale di test sensore 1 - 16 bit con segno
6,7	Segnale di test sensore 2 - 16 bit con segno
8,9	Test interno- Valore di calibrazione 1 - 16 bit con segno
10,11	Test interno – Valore di calibrazione 2 - 16 bit con segno
12	Inutilizzato- Massa
13	Inutilizzato - Massa
14	Flag di calibrazione eseguita – Inutilizzato
15	Comando di calibrazione – Inviare 255 per entrare nella modalità calibrazione e 0 per uscire

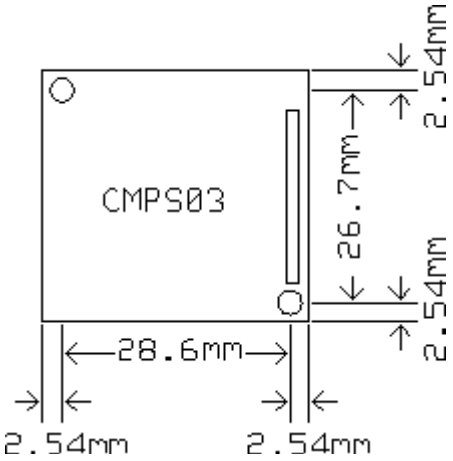
IMPORTANTE Durante la calibrazione la bussola deve essere tenuta in piano (orizzontale e parallela alla superficie terrestre) con la faccia su cui sono montati i componenti girata verso l'alto. Tenere il circuito lontano da oggetti metallici, in particolare se sono magnetici .

Quando si effettua la calibrazione si devono conoscere esattamente le direzioni in cui si trovano i quattro punti cardinali verificandole con una bussola magnetica.
Per calibrare la bussola bisogna portare il livello del pin 6 prima a livello basso e poi a livello alto per ciascuno dei punti cardinali (in un ordine qualunque a partire dal nord): questo può essere fatto facilmente collegando un pulsante tra il pin stesso e massa.

Ad esempio:
1 .Mettere il circuito in piano diretto verso il nord. Premere e rilasciare il pulsante
2. .Mettere il circuito in piano diretto verso il nord. Premere e rilasciare il pulsante
3. .Mettere il circuito in piano diretto verso il nord. Premere e rilasciare il pulsante
4. .Mettere il circuito in piano diretto verso il nord. Premere e rilasciare il pulsante
Calibrazione terminata.

QUOTATURA PCB

Il seguente disegno mostra le posizioni dei fori di montaggio del CMPS03 PCB.



Per informazioni aggiuntive sulla calibrazione:
http://www.robot-italy.net/downloads/cmps03_calibration.PDF

Per interfacciare la bussola con il bx-24 tramite protocollo I2C:
http://www.robot-italy.net/downloads/cmps03_bx24app.pdf

Vai a Robot “Pompiere”
Vai a Robot “Guardiano”
Vai a Possibili sviluppi
Torna all’indice

CMUCAM2 E CMUCAM2HR E SUPPORTO SU DUE ASSI

La CMU cam è incorpora un frame-buffer il quale consente una buona flessibilita' nella manipolazione dell'immagine, sottocampionamento e un maggiore frame rate. E' equipaggiata con un microcontrollore Scenix SX52 interfacciato con una camera CMOS OV6620 (OV7620 nella HR) della Omnivision.
La CMU cam II comunica via porta seriale RS-232 o TTL.
Caratteristiche:

- Individua oggetti di colore predefinito fino ad una frequenza di 50 fotogrammi/secondo
- Individua il movimento fino a 26 fotogrammi/secondo
- Trova il centro di ogni dato individuato
- Restituisce un istogramma a 28 bin per ogni canale di colore
- Manipolazione di immagini differenziate per pixel orizzontali
- Trasferisce una bitmap binaria in tempo reale dei pixel tracciati
- Proprietà dell'immagine acquisita impostabili
- Risoluzione fino a 143x80 (l'HR con il modulo OV7620 arriva a risoluzione 255 x 176)
- Uscita B/W analogica (PAL o NTSC)
- Modalita' a basso consumo
- Supporta velocita' multiple ((115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200)
- Controlla 5 uscite per servi R/C tramite i servo outputs
- Controlla automaticamente 2 servi per individuare i colori su 2 assi
- Uscita del segnale personalizzabile

- Processa una immagine bufferizzata a passaggi multipli

Uno degli usi principali della CMUcam2 è quello di localizzare e monitorare i colori. Le migliori prestazioni si ottengono quando ci sono colori intensi e con elevato contrasto. Per esempio si può facilmente individuare una palla rossa su uno sfondo bianco ma sarebbe difficile distinguere diverse tonalità di uno stesso colore. L'individuazione di oggetti colorati può essere utilizzata per localizzare punti di riferimento e seguirli. Se vengono rilevati cambi netti di colore allora è molto probabile che qualcosa nella scena sia cambiato. Nella modalità "line mode" la CMUcam2 può ottenere in modo semplice immagini a colori binarie in bassa risoluzione. Questa modalità può essere utilizzata per eseguire l'inseguimento di linee in maniera più complessa, ad esempio rilevando forme e variazioni di colori decise. Queste operazioni più avanzate richiedono algoritmi appositi per il post processing delle immagini binarie inviate dalla CMUcam2.

COME INDIVIDUARE UN OGGETTO

Per specificare un colore bisogna definire un valore massimo e un valore minimo per ognuno dei tre canali del colore (RGB).

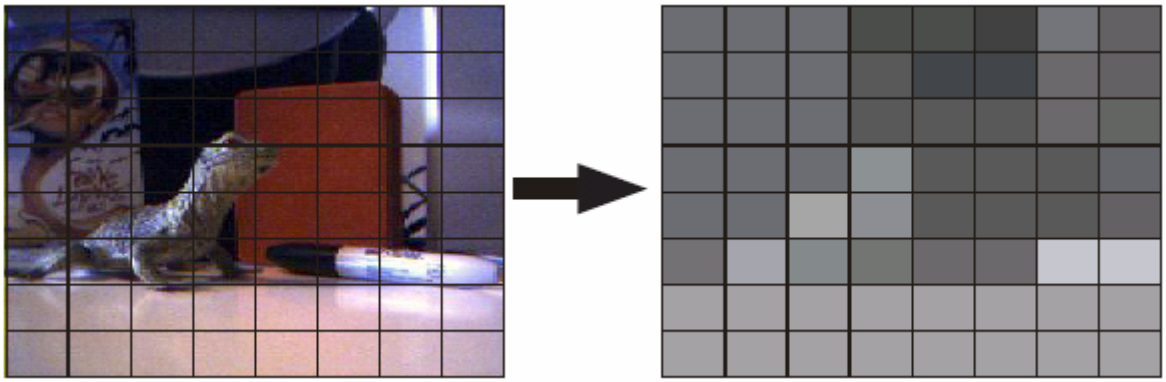
Ogni colore è rappresentato da un valore per ognuna delle componenti (Rosso, Verde, Blu) che indica quanto di quel canale è miscelato nel colore finale. Nel caso della CMUcam2 ogni canale di colore è convertito in un valore compreso tra 16 e 240.

Dopo aver individuato i limiti l'immagine viene processata secondo il seguente algoritmo: per ciascun frame viene esaminato ogni pixel, riga per riga a partire dall'angolo in alto a sinistra dell'immagine, quindi si verifica se questi ultimi ricadono all'interno dell'intervallo di colore definito creando progressivamente una regione (Bounding Box) con caratteristiche di colore simili. Viene inoltre rilevata la posizione del pixel esaminato in quel momento per controllare se è più a sinistra, più a destra, più in basso o più in alto di tutti gli altri pixel

rientranti nel range stabilito ed individuati fino a quel momento. Se il pixel si trova all'esterno dei confini della regione già identificata, la stessa verrà estesa fino a comprendere nuovi pixel.
Successivamente con un apposito algoritmo viene calcolato il baricentro dell'area rilevata; questa informazione insieme alla bounding box può essere utilizzata per calcolare la confidenza dell'oggetto tracciato.

IDENTIFICAZIONE DELLE VARIAZIONI TRA DIVERSI FRAME

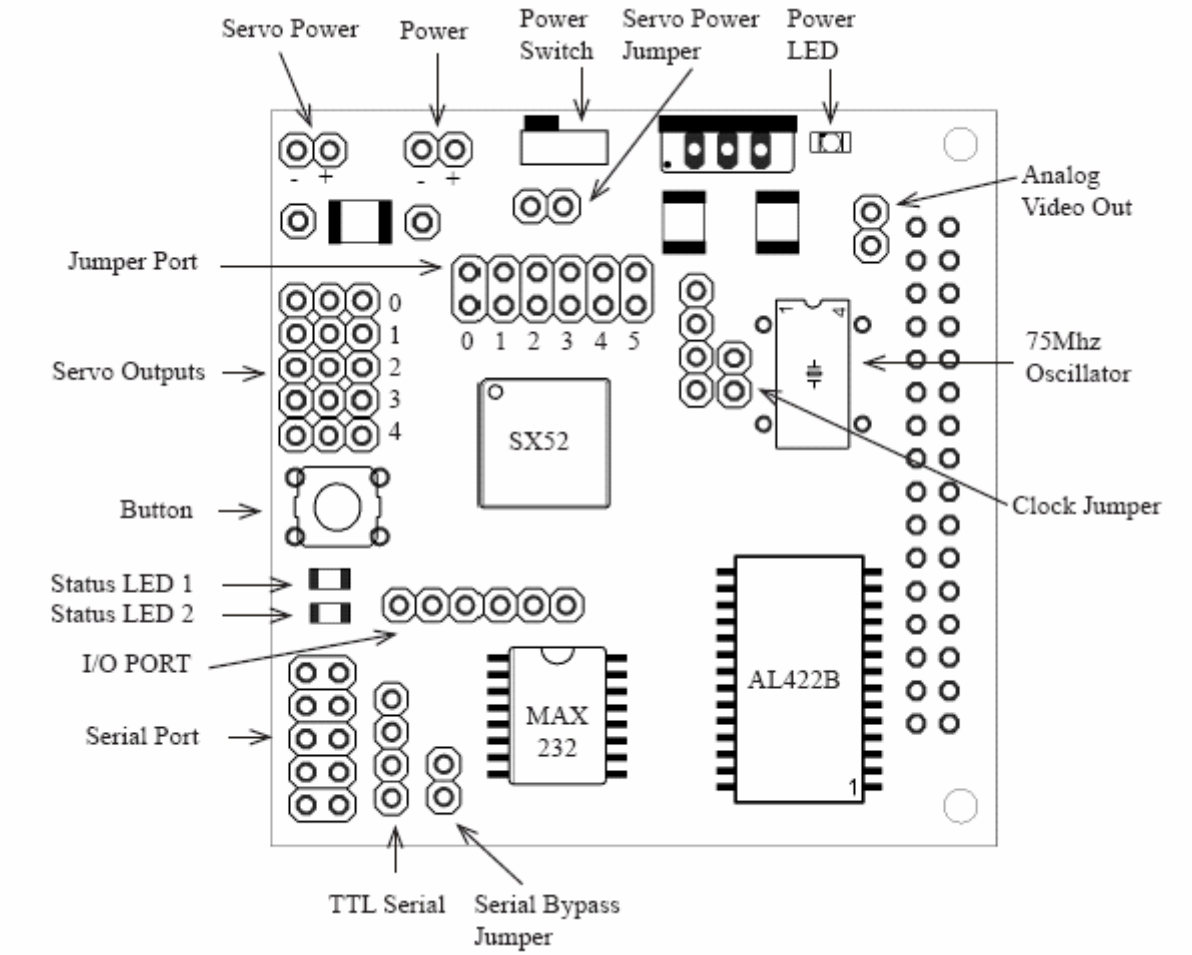
Per identificare i cambiamenti in una serie di immagini (dovuti ad esempio al movimento di un oggetto) la CMUcam2 utilizza una funzione di differenziazione dei frames. Ogni immagine viene suddivisa in 64 blocchi, a ciascuno dei quali viene associato un byte che rappresenta l'informazione legata a quel blocco. Quando una nuova immagine viene immagazzinata viene anch'essa suddivisa in una matrice di blocchi 8x8, in modo da confrontarla con quella precedente così da rilevare eventuali cambiamenti nell'informazione e quindi nell'immagine stessa. Questa funzione può risultare utile, nel caso che la telecamera sia montata su un supporto semovibile, per seguire un oggetto.



COLLEGAMENTO E SETTAGGIO HARDWARE

Per testare la CMUcam2 sono necessari un cavo seriale, l'alimentazione e un computer dotato di porta seriale. La tensione di alimentazione applicabile può variare tra 6 e15 V DC; l'assorbimento di corrente è di circa 200mA.

PROCEDURA DEL TEST:



1) Connettere l'alimentazione al circuito rispettando la polarità

2) Connettere il cavo seriale tra la videocamera e il computer: questo collegamento è richiesto inizialmente per testare e mettere a fuoco (regolando la ghiera situata sull'obiettivo) la camera

3) Assicurarsi che il jumper di clock sia connesso (necessario per pilotare il processore)

4) Una volta che tutto è stato collegato provare ad accendere il circuito: a questo punto dovrebbero illuminarsi il LED di alimentazione ed i due LED di stato , di cui solo uno dovrebbe rimanere acceso a segnalare la corretta configurazione della videocamera.

MESSA A FUOCO DELLA CMUCAM2 ATTRAVERSO GRAPHICAL USER INTERFACE(GUI)

Al primo avvio della camera la lente molto probabilmente non sarà a fuoco. Per mettere a fuoco la telecamera bisogna osservare alcune delle immagini acquisite. Il modo più semplice per farlo è utilizzare un interfaccia grafica che può visualizzare i pacchetti di

immagini. Una possibilità è quella di utilizzare la CMUcam2GUI,un programma java che si può scaricare dal sito http://www.robot-italy.net/downloads/CMUcam2GUI.zip

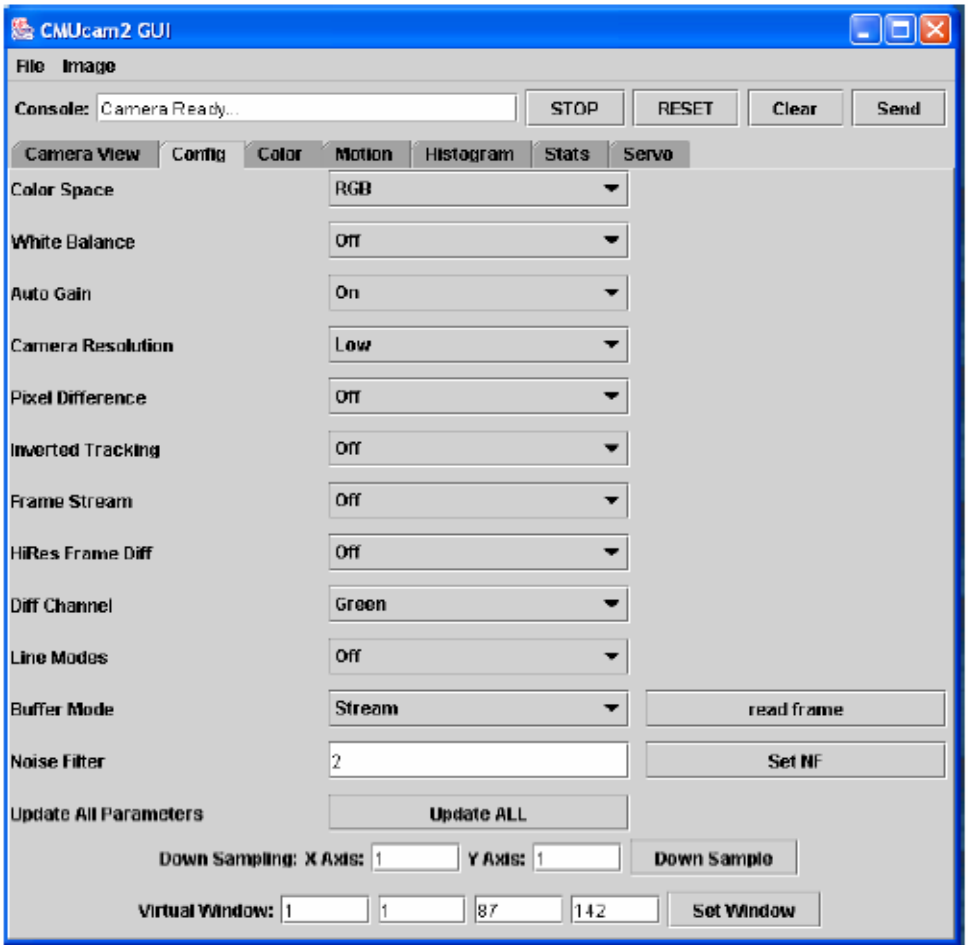
VERIFICARE DI AVERE JAVA INSTALLATO SULCOMPUTER

Il modo più semplice per verificare se si ha già java installato sul proprio computer è quello di andare nel menu "start" di windows, cliccare su esegui e poi digitare "command" nella finestra di dialogo apparsa per far apparire il prompt del dos. A questo punto digitare "Java-version" nella linea di comando; se java è installato compare la versione del software altrimenti verrà visualizzato un messaggio d'errore. In quest'ultimo caso bisognerà scaricarne una copia dal seguente link: http://www.java.sun.com. Assicurarsi che la versione scaricata sia la 1.4.0 o superiore.

AVVIARE LA CMUCAM2GUI

Dopo aver installato java,scaricare una copia dell'ultima versione del programma CMUcam2GUI.Estrarre i file dall'archivio CMUcam2GUI.zip e cliccare due volte sul file jar CMUcam2GUI.

PANNELLO DI CONFIGURAZIONE DELLA GUI

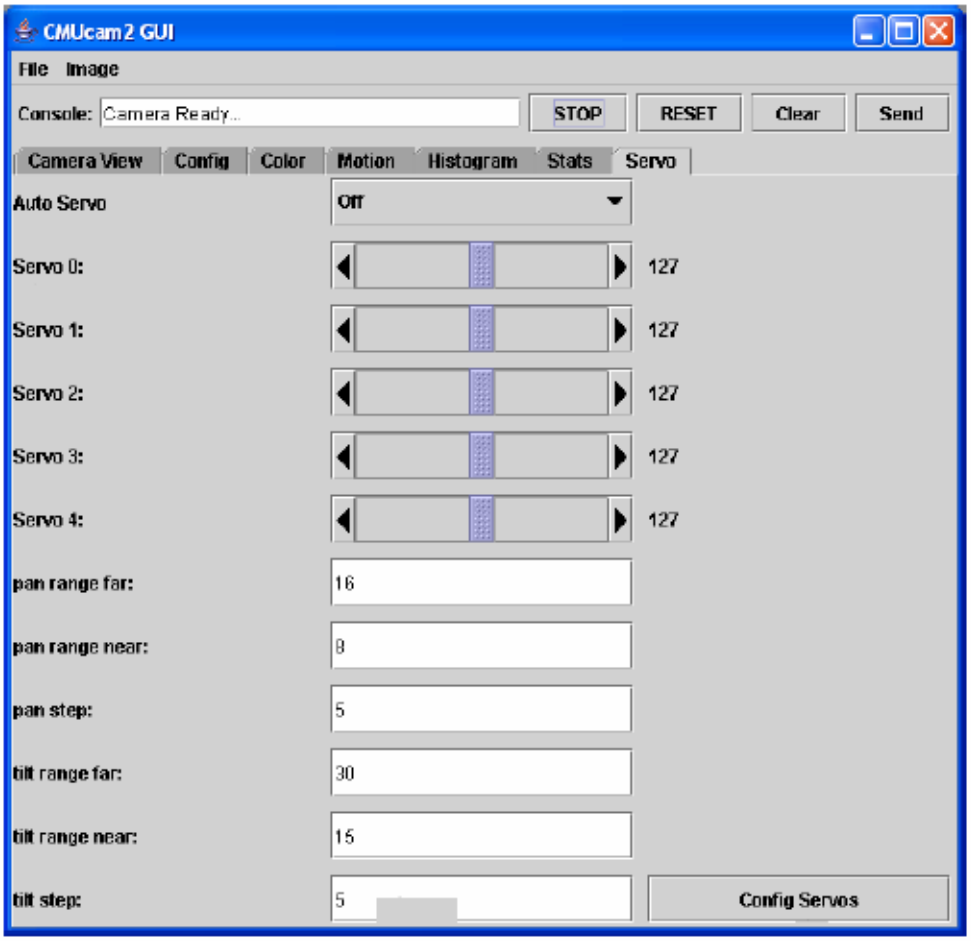


Questo pannello consente di inviare alcuni comandi modificando quindi i parametri della CMUcam2 eccetto quelli dei servocomandi. Ogni menù a tendina richiama uno dei comandi di configurazione della camera che è possibile modificare a seconda delle esigenze. Questo semplifica le operazioni di controllo della camera, infatti i comandi inviati dalla GUI sono equivalenti a quelli che possono essere inviati programmando direttamente il microprocessore.

PANNELLO PER IL CONTROLLO DEI SERVI TRAMITE GUI

Questo pannello permette di testare e configurare i servi: le barre a scorrimento consentono di variare semplicemente la posizione dei servi senza utilizzare l'istruzione software SV (Servo Position). Il range operativo del servo, di default, è compreso tra 46 e 210 e la posizione centrale è indicata dal valore 128 a cui corrisponde la generazione di in impulso della durata di 1,5mS.

La parte sotto invece serve a modificare i parametri del funzionamento automatico dei servocomandi (istruzione software SP)



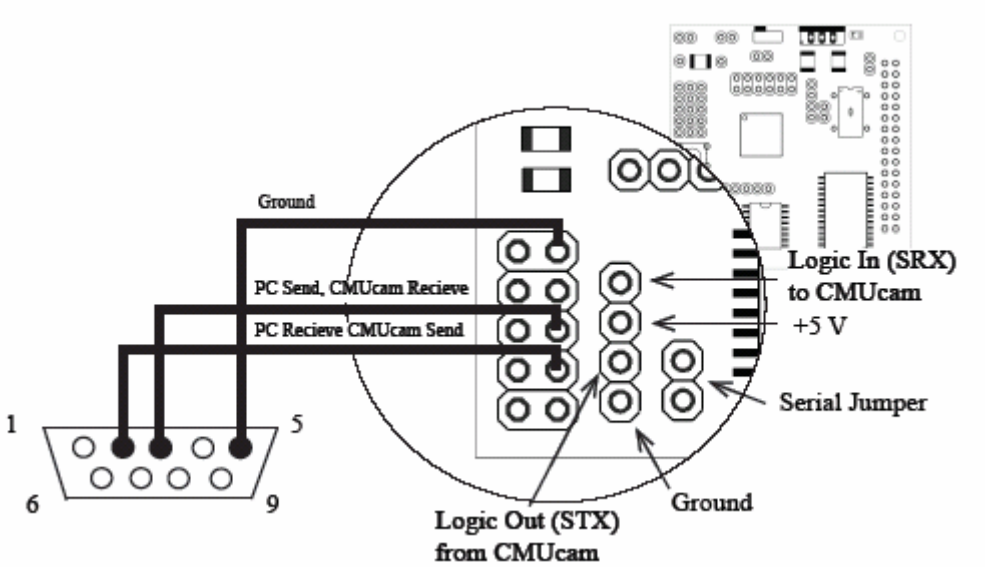
DEMO MODE

Il DEMO MODE fa aprire la finestra di tracciamento della GUI (Graphic User Interface) e pilota i due servi standard verso l'oggetto da tracciare (questo può essere inizializzato autonomamente all'avvio). Per prima cosa bisogna collegare i due servi, che controllano la rotazione e l'inclinazione della videocamera, rispettivamente alle porte servo 0 e 1 e assicurarsi poi che essi siano correttamente alimentati. Tenendo premuto il bottone push accendere la videocamera. A questo punto il LED di tracciamento dovrebbe cominciare a lampeggiare rapidamente; rilasciare quindi il bottone e aspettare che il LED smetta di lampeggiare. Successivamente puntare la videocamera verso un oggetto colorato e premere nuovamente il bottone "push". Questa procedura permette alla telecamera di registrare il colore dell'oggetto e ai servi di cominciare a spostare automaticamente la stessa verso l'oggetto. Se i servi si muovono nella direzione opposta a quella desiderata aggiungere l'apposito jumper per l'inversione della direzione dei servi.

E' possibile consultare il manuale completo al seguente link:

http://www.robot-italy.net/downloads/CMUcam2_manual.pdf
Manuale interfaccia grafica (inglese):
http://www.robot-italy.net/downloads/CMUcam2GUI_overview.pdf

PORTA SERIALE



Per collegare la scheda alla porta seriale del PC vengono utilizzati 3 dei 10 pin a disposizione sulla seriale della scheda: uno per inviare dati, uno per riceverli e la massa. In alternativa alla seriale per comunicare con un microcontrollore si può utilizzare il connettore TTL rimuovendo il serial jumper.

PORTA JUMPER

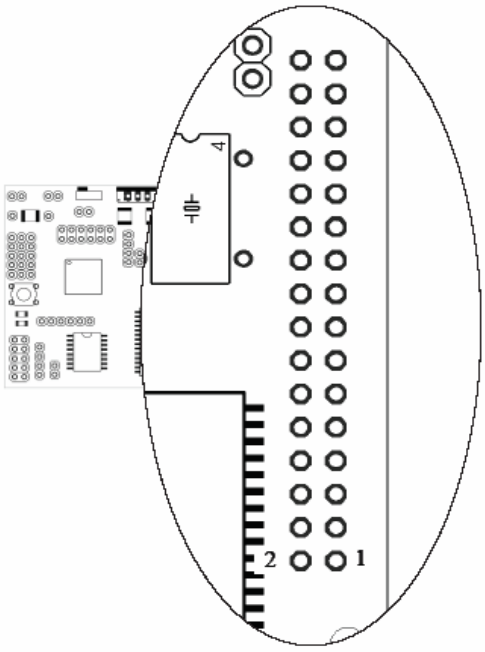
Per settare diverse velocità di comunicazione tra scheda e PC si possono utilizzare i jumper 3, 4 e 5, mentre i jumper 0 e 1 servono per invertire la direzione di movimento dei servi. Il jumper 3, detto di slave mode, serve nel caso che si vogliano utilizzare più schede per processare i dati provenienti dalla videocamera.

Baud Rate	Pin 3 4 5
115,200 Baud	_ _ _
57,600 Baud	_ _ X
38,400 Baud	_ X _
19,200 Baud	_ X X
9,600 Baud	X _ _
4,800 Baud	X _ X
2,400 Baud	X X _
1,200 Baud	X X X

X - jumper closed
_ - jumper open

BUS DELLA TELECAMERA

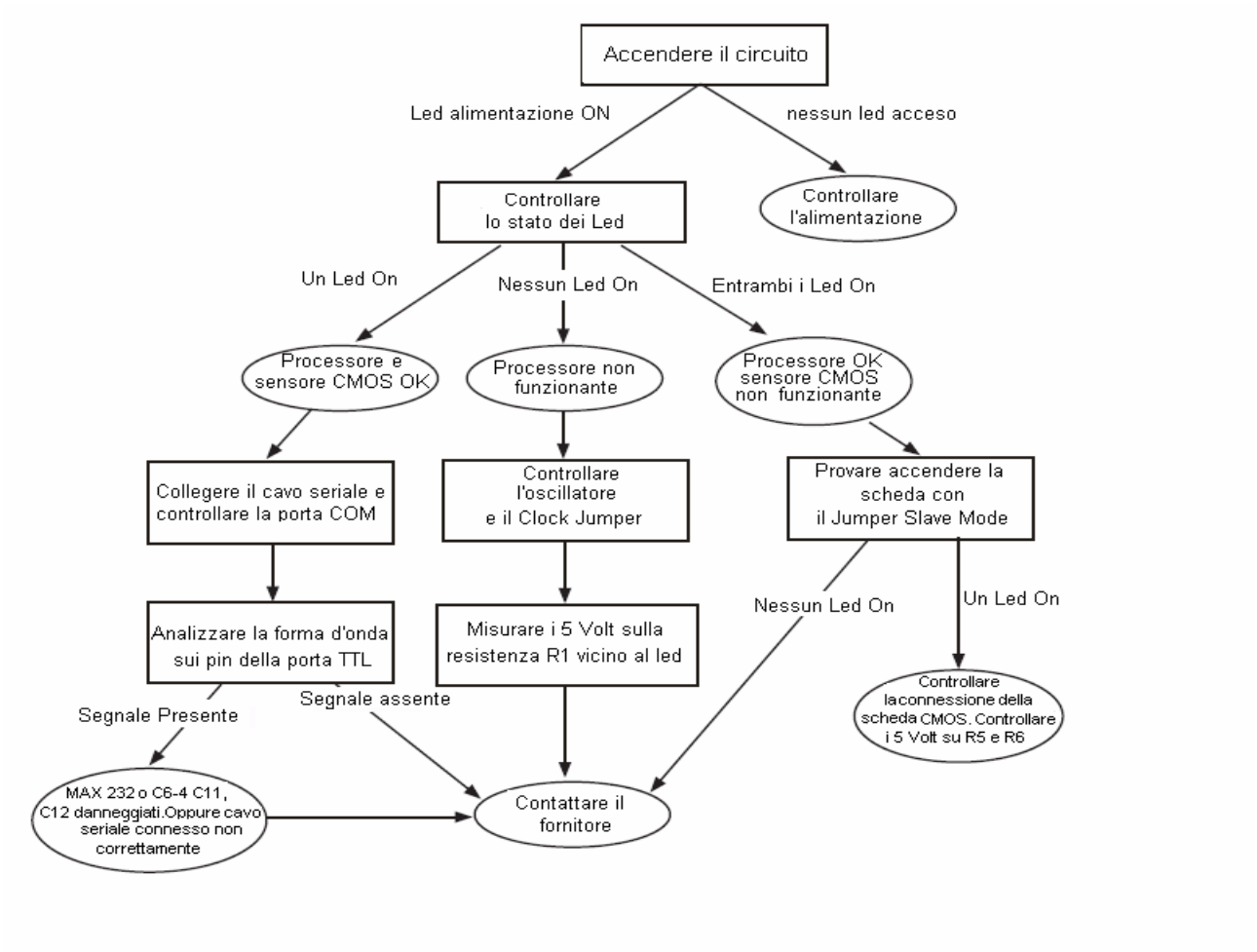
Se si vuole utilizzare la telecamera senza montarla direttamente sulla scheda si può utilizzare un cavo flat a 32 pin. A seconda dell'uso che si fa del sensore CMOS alcuni pin possono essere lasciati scollegati: ad esempio se il sensore CMOS è alimentato direttamente dal supporto su cui è montato si possono lasciare scollegati i pin dell'alimentazione.



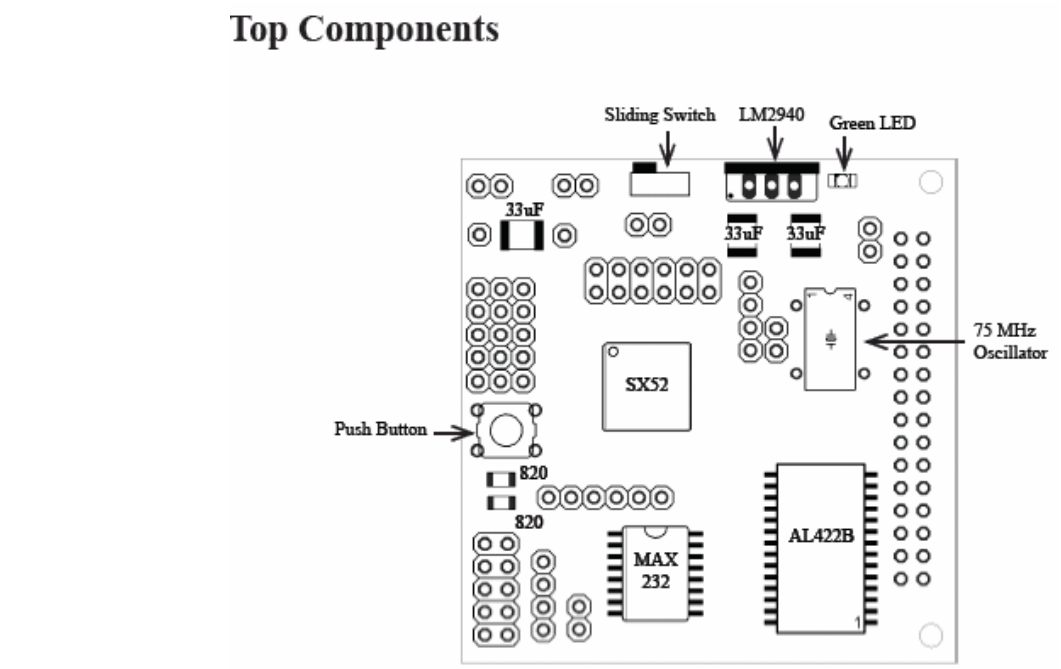
1-8	Y0-Y7	Digital Output Y Bus
9	PWDN	Power Down Mode
10	RST	Reset
11	SDA	I2C Serial Data
12	FODD	Odd Field Flag
13	SCL	I2C Serial Clock
14	HREF	Horizontal Ref
15	AGND	Analog Ground
16	VSYNC	Vertical Sync
17	AGND	Analog Ground
18	PCLK	Pixel Clock
19	EXCLK	External Clock
20	VCC	+5 VDC
21	AGND	Analog Ground
22	VCC	+5 VDC
23-30	UV0-UV7	Digital Output UV BUS
31	GND	Common Ground
32	VTO	Video Out (75Ohm)

RISOLUZIONE PROBLEMI

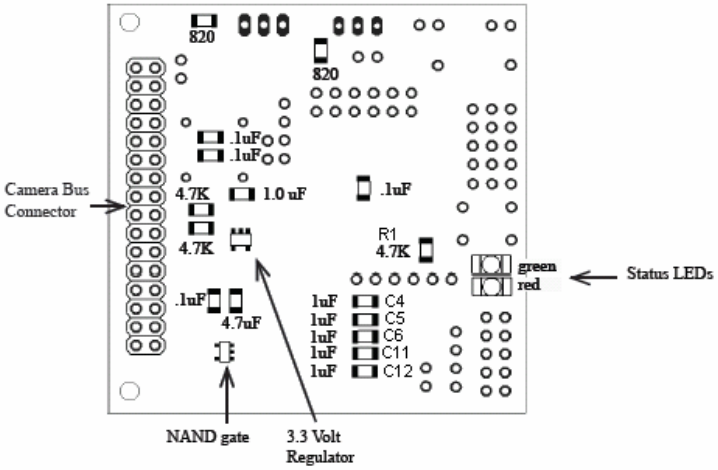
Il diagramma nella figura successiva può essere utile per diagnosticare i problemi hardware della CMUcam2.


Accendere il circuito
Led alimentazione ON
nessun led acceso
Controllare lo stato dei Led
Controllare l'alimentazione
Un Led On
Nessun Led On
Entrambi i Led On
Processore e sensore CMOS OK
Processore non funzionante
Processore OK sensore CMOS non funzionante
Collegere il cavo seriale e controllare la porta COM
Controllare l'oscillatore e il Clock Jumper
Provare accendere la scheda con il Jumper Slave Mode
Analizzare la forma d'onda sui pin della porta TTL
Misurare i 5 Volt sulla resistenza R1 vicino al led
Nessun Led On
Un Led On
Segnale Presente
Segnale assente
Controllare la connessione della scheda CMOS. Controllare i 5 Volt su R5 e R6
MAX 232 o C6-4 C11, C12 danneggiati.Oppure cavo seriale connesso non correttamente
Contattare il fornitore

CABLAGGIO DEI COMPONENTI

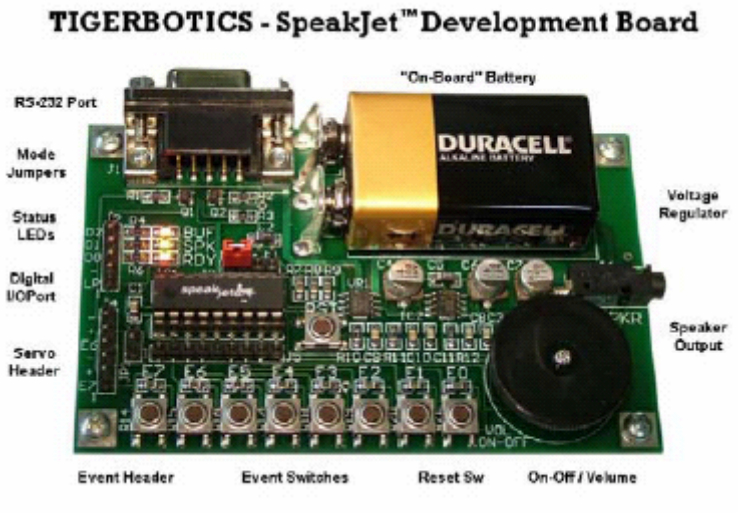


Bottom Components



Vai a Robot “Pompiere”
Vai a Robot “Guardiano”
Vai a Possibili sviluppi
Torna all’indice

SINTETIZZATORE VOCALE (SpeakJet)



La scheda di sintesi vocale Tigerbotics (SpeakJet) è progettata specificatamente per supportare il chip di sintesi vocale e sonora Magnevation "SpeakJet". La circuiteria sulla scheda permette all'utente di utilizzare il chip sia manualmente, tramite pulsanti e LED, sia in remoto, connettendo le porte del chip a dei controlli esterni o ad un microprocessore.

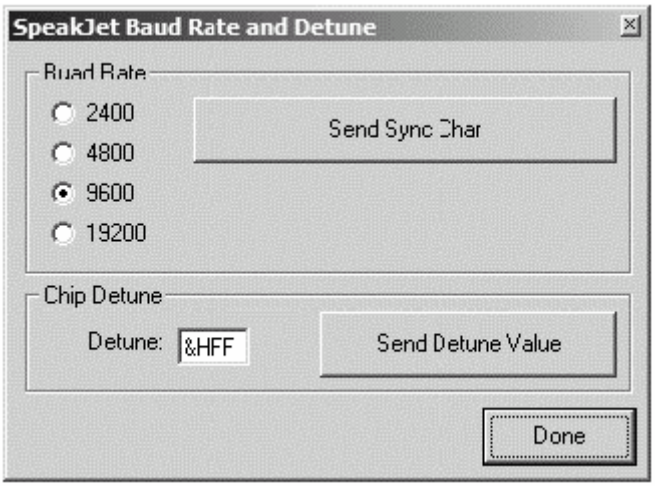
COMPONENTI NECESSARIE ALL'INSTALLAZIONE

1)PC con Windows 98 o versioni più recenti.
2)Ultima versione del software "PhraseALator" (v1.4 o più recente) per gestire il chip scaricabile a questo indirizzo: www.tigerbotics.com/support/speakjet/.
3)Cavo seriale standard DB-9 (connettore maschio ad un capo femmina all'altro) da collegare tra PC e connettore J1 sulla scheda.
4)Pila a 9V per alimentare la scheda.
5)Piccolo altoparlante con impedenza da 8 o 16Ω e cavo con spina da 3,5mm.

CONFIGURAZIONE SOFTWARE

Nella schermata iniziale (vedi figura sotto) assicurarsi che l'opzione "Use Flow Control" non sia selezionata e che il numero della Porta Seriale sia corretto: se il computer utilizza la COM1 allora dovrebbe esserci il numero 1 nella finestra "Serial Port".

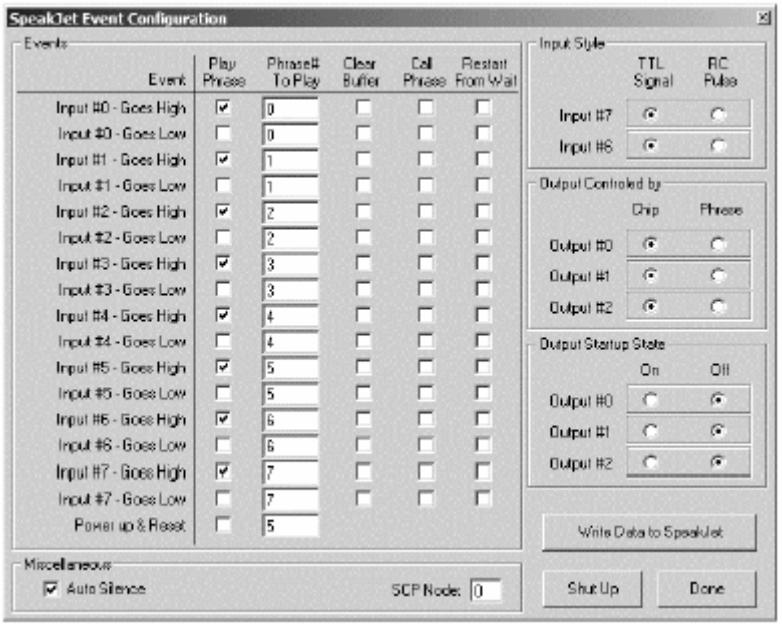
Assumendo che il computer, l'alimentazione e l'altoparlante siano connessi, accendere la scheda ruotando di mezzo giro il controllo volume. Cliccare quindi su "Baud Rate and Detune" e selezionare "9600" come mostrato nella figura:



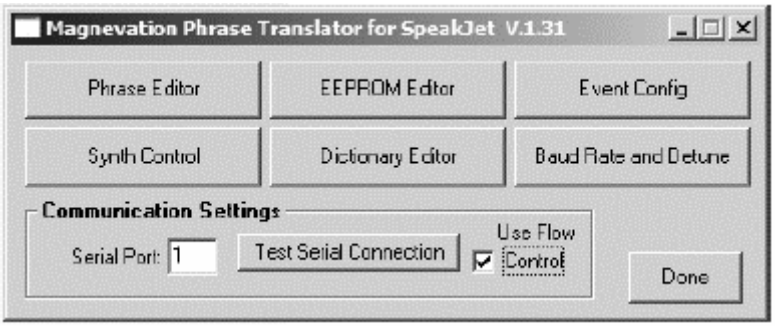
Verifica della comunicazione tra il computer ed il sintetizzatore vocale: impostare il sintetizzatore in modalità Configurazione del Baud Rate(per entrare in tale modalità spostare il jumper J3 dalla posizione "M0" a "M1"); a questo punto probabilmente si sentirà il sintetizzatore entrare in modalità "Demo". Premere quindi il pulsante "RESET" (RST) una volta: un beep intermittente emesso dall'altoparlante indica che si è entrati nella modalità Configurazione del Baud Rate. Prima di procedere riposizionare il jumper in posizione "M0". Ora sullo schermo del computer cliccare su "Send Sync Char" (SSC) e il beep dovrebbe fermarsi ad indicare che il computer ed il sintetizzatore sono completamente sincronizzati. Cliccare su "Done" per ritornare al menu principale. Se il beep non si è fermato provare a cliccare nuovamente su SSC. Se ancora il segnale acustico non è terminato probabilmente avete selezionato il numero di porta seriale (COM) errato nel menu principale o avete un problema con il cavo.

Ora che si è in comunicazione con la scheda bisogna assicurarsi che l'integrato sia settato per il normale funzionamento. Cliccare su "Event Config" e configurare la schermata come

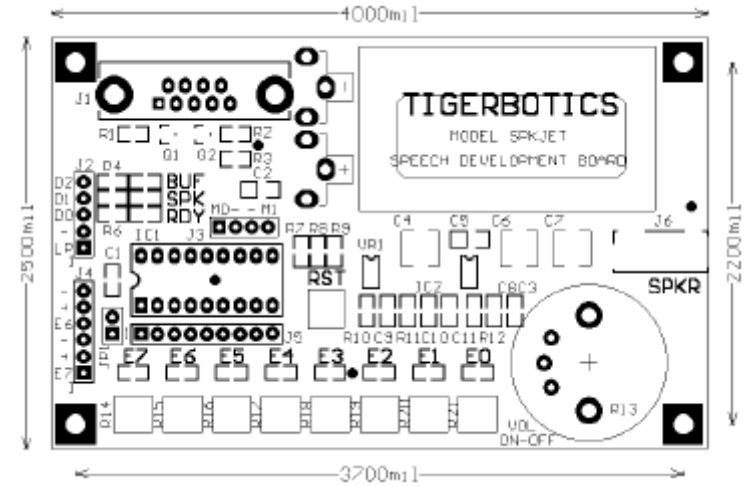
mostrato nella figura sotto. Una volta terminato, cliccare su “Write Data to SpeakJet” per inviare la configurazione al chip e quindi su “Done”per tornare al menu principale.



Con il chip predisposto per la comunicazione e correttamente impostato si può continuare con l’ultimo passo del processo di installazione. Bisogna attivare il segnale “Flow Control” tra l’integrato ed il computer. “Flow Control” è una linea di controllo della comunicazione che indica al computer di non inviare dati quando il chip è troppo impegnato per riceverli. Questa linea previene la perdita di dati durante download lunghi. Attivare “Flow Control” evidenziandolo come mostrato in figura.



CABLAGGIO



I connettori che permettono di collegarsi alle linee per il monitoraggio ed il controllo esterno sono indicati come J2,J4 e J5. I segnali disponibili tramite i connettori sono:

J2 – Connettore Uscita Digitale

Pin #1 "LP" LED di Alimentazione (vedere note)
Pin #2 "-" Massa
Pin #3 "D0" Output Dati "0" / PRONTO
Pin #4 "D1" Output Dati "1" / ALTOPARLANTE IN FUNZIONE
Pin #5 "D2" Output Dati "2" / BUFFER PIENO

J2 Note: Il pin "LP" (#1) su J2 è connesso alla massa (Pin #2) sul lato inferiore della scheda. Se in qualche applicazione ci fosse necessità di disabilitare le luci LED sulla scheda per ridurre la potenza o il carico su una linea di segnale, si può tagliare la pista tra i pin #1 e #2 sul lato inferiore della scheda. Nel caso si vogliano riconnettere le linee è possibile unire i pin #1 e #2 con un jumper.

J4 – Ingressi Impulsi Servi

Pin #1 "E7" Evento #7 / RC1 (segnale servo)
Pin #2 "+" Opz. +5v (vedere note)
Pin #3 "-" Massa
Pin #4 "E6" Evento #6 / RC0 (segnale servo)
Pin #5 "+" Opz. +5v (vedere note)
Pin #6 "-" Massa

J4 Note: Il connettore J4 fornisce accesso ai due "Ingressi di impulso" del chip. Il connettore è configurato in gruppi da tre pin come i connettori servo standard in modo da poter collegare un normale servo RC. I pin #2 e #5 possono essere connessi opzionalmente all'alimentazione a +5V regolati inserendo un jumper su JP1. Il consumo di corrente da

questi pin deve essere limitato a 25mA (totale). Questi pin sono stati progettati per fornire segnali di controllo più che per alimentare dispostivi esterni. ***ATTENZIONE:*** Non attivare mai i +5V con JP1 se si utilizza questa porta con un ricevitore radio o qualsiasi dispositivo che possa fornire potenza a questo pin.

J5 – Input di Innesco di Evento

Pin #1 "E7" Evento #7 / RC1
Pin #2 "E6" Evento #6 / RC0
Pin #3 "E5" Evento #5
Pin #4 "E4" Evento #4
Pin #5 "-" Alimentazone di massa
Pin #6 "E3" Evento #3
Pin #7 "E2" Evento #2
Pin #8 "E1" Event #1
Pin #9 "E0" Event #0

J5 Note5: Gli "Ingressi di Evento" sono collegati internamente a +5volt tramite un resistore da 10KΩ in modo che per innescare un evento le linee debbano essere solo portate a livello basso (per far ciò la massa è disponibile sul pin #5 del connettore).Le "linee di Evento" possono essere attivate anche dai "Pulsanti" E0-E7.Nel nostro robot vengono portati a livello logico basso attraverso il bx-24 sul Mini-ABB

FONTI ESTERNE DI ALIMENTAZIONE

Se non si vuole alimentare la scheda con una batteria da 9V, la si può collegare (sempre utilizzando lo stesso connettore) ad un alimentatore esterno che deve fornire una tensione compresa tra 6 e 12V (MAX). ATTENZIONE: se si utilizza un connettore che abbia già i fili installati (solitamente ROSSO e NERO ad indicare la polarità), LA POLARITA' SARA' SBAGLIATA! Questo accade perchè il connettore è pensato per essere collegato ad una batteria e non ad un altro dispositivo. Assicurarsi che la polarità sia corretta prima di applicare l'alimentazione o si potrebbero danneggiare la scheda o il chip.

Manuale in inglese:
http://www.tigerbotics.com/files/speakjetman090904.pdf

Vai a Robot "Pompiere"
Vai a Robot "Guardiano"
Vai a Possibili sviluppi
Torna all'indice